The Journal of Systems & Software 204 (2023) 111800

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Agile software engineers' affective states, their performance and software quality: A systematic mapping review☆

Martín G. Salido O., Gilberto Borrego, Ramón René Palacio Cinco, Luis-Felipe Rodríguez *

Departamento de Computación y diseño, Instituto Tecnológico de Sonora, Mexico

ARTICLE INFO

Article history:
Received 19 July 2022
Received in revised form 25 May 2023
Accepted 3 July 2023
Available online 6 July 2023

Keywords:
Affective state
Agile methodology
Software Engineering
Software Engineers' performance
Software quality

ABSTRACT

Nowadays, software development companies use agile methodologies to increase the speed and flexibility required by unpredictable working environments and streamline the software delivery process. Agile methodologies emphasize the software engineers' interactions rather than rigid software development processes. Therefore, agile software development processes (e.g., implementing user stories) should consider software engineers' affective states since these influence agile software project activities (e.g., team composition or decision-making). We conducted a systematic mapping review following the guidelines of Petersen, K. and Kitchenham, B. to answer our research question: "What affective states influence agile software engineers' performance and developed software quality?". We retrieved over 16,000 articles published between January 2010 and September 2021 and after applying selection criteria, 24 primary articles were identified. The results show that the affective states of software engineers influence the activities of the software project and the software development process. Furthermore, we found that there is a lack of well-defined and standard metrics to study the influence of software engineers' affective states on their performance and the quality of the resulting software. Finally, we concluded that studying and understanding the affective states of software engineers in agile environments is crucial to achieve their well-being at work and improve their performance.

1. Introduction

Nowadays, software industries adopt the agile software development (ASD) paradigm to enable software teams to rapidly deliver high-quality software to their customers (work in small but consumable) (digital.ai, 2021; Hayat et al., 2019). The *heart of agile* (Cockburn, 2016) expresses that personal and emotional information about the team, along with the state-of-the-art techniques in psychotherapy and family coaching compatibles with agile development, have to be taken into account by agile software engineers to learn how to work and collaborate to do what they need to do.

Methodologies based on the ASD paradigm take the agile manifesto as their core; they value the talent, knowledge, and attitude of individuals and their interaction (e.g., relationship with the client or team members) over rigid processes for software development (Beck et al., 2001; Uribe and Ayala, 2007). In fact, the fourth and the fifth principle of the agile manifesto emphasize the value of communication, teamwork, and quality of the work environment to improve the software projects' management (Pedrycz et al., 2011; digital.ai (formerly CollabNet VersionOne), 2020). The twelfth principle (Beck et al., 2001; Fowler et al., 2001) declares that agile teams should continue changing whatever is necessary to improve their performance and adjust to starting over if there are plan deviations. The agile paradigm shifted the focus from significant initiatives to small teams, from big promises to incremental deliveries, and from hand-offs between specialists to collaboration and self-sufficiency (Lander et al., 2016). In addition, Cockburn and Highsmith (2001) mentioned that agile development focuses on the talents and skills of individuals, molding the process to specific people and teams.

Agile Software Development focuses on communication and customer involvement, which implies that team interpersonal and social skills are crucial to delivering efficient modules to customers in each iteration (Petersen and Wohlin, 2009). Moreover, in the agile paradigm, software team members frequently iterate between the different phases of systems development (e.g., planning, designing, coding, and testing) and change their roles accordingly (Lindsjørn et al., 2016; Balijepally et al., 2006), which facilitates communication and collaboration among them (Cockburn and Highsmith, 2001). Besides, agile methods are more sensitive to human factors (e.g., affective states), relying heavily

☆ Editor: Xiao Liu.

* Corresponding author.
E-mail addresses: marting.salido@potros.itson.edu.mx (M.G. Salido O.), gilberto.borrego@itson.edu.mx (G. Borrego), ramon.palacio@itson.edu.mx (R.R. Palacio Cinco), luis.rodriguez@itson.edu.mx (L.-F. Rodríguez).

https://doi.org/10.1016/j.jss.2023.111800
0164-1212/

on improving the software engineers' interpersonal communications (Stamelos and Sfetsos, 2007); hence, when the developers' and customers' interpersonal relationships are fostered in an agile context, they can benefit the software outcome (Madsen and Matook, 2010).

Unlike the agile paradigm, traditional software development depends on linear processes where each software developer has a specific role and creates an output (commonly a written document) that is used as the input for the next phase (Diebold and Dahlem, 2014; Petersen and Wohlin, 2009). Hence, the perception of the importance of human aspects in software development, e.g., ''individuals and interactions over processes and tools'', led to the creation of the agile paradigm (Beck et al., 2001; Uribe and Ayala, 2007; Graziotin et al., 2014a), where development is a people-oriented process that focuses on the role of human factors, such as the team and their interactions (Balijepally et al., 2006).

Affective states (e.g., emotions, moods, and feelings) are factors that influence software engineers' performance and software quality regardless of the software development environment (Fagerholm and Münch, 2012). Therefore, the software development can be considered an intrinsically human-based intellectual activity performed through cognitive processing tasks (Fagerholm and Münch, 2012; Graziotin et al., 2015). Affective states, such as emotions, are one of the decisive factors that influence these individuals' cognitive processes (e.g., learning, reasoning, or decision-making) at the social and psycho-motor level and even their health (Ekman and Davidson, 1994; Rodríguez et al., 2011). In fact, individuals react to perceived entities and events emotionally, even when they interact with computers, which are commonly considered purely logical and mechanical devices (Landowska, 2013). Hence, software engineers' affective states heavily influence processes related to their performance (e.g., problem understanding, or teamwork) and software quality (Graziotin et al., 2014b; Hedberg Giffith and Nguyen, 2018; Girardi et al., 2019; Murgia et al., 2014).

Although the agile manifesto emphasizes the importance of people's hopes, emotions, ambitions, and grievances, it does not provide details for taking these factors into account in the software development and their implications (Nagappan, 2021). According to Girardi et al. (2018), the risk of developers' emotional and mental exhaustion could increase when companies that promote and apply agile software development do not apply the agile principles correctly or when their developers do not align with those principles. In addition, Cao and Park (2017) stated that emotional experiences of software engineers can play a significant role in the success of agile projects since emotions can motivate or intensify the engineers' behavior. This situation causes rejection to significant events (e.g., making changes in the code) or affects the decision-making process during agile development (Cao and Park, 2017). For example, when software engineers experience frustration, stress, or rage at facing obstacles, such as the development of complex or wide-ranging projects, it may affect engineers' capacity to regulate their affective states (e.g., emotions), bringing, as a consequence, a decrease in their performance (Graziotin et al., 2018; Crawford et al., 2014; Bishop et al., 2020; Fountaine and Sharif, 2017; Novielli et al., 2018; Graziotin et al., 2014a). In contrast, happier software engineers are better at solving analytical problems (Graziotin et al., 2018). Therefore, factors such as the emotional experiences, well-being, and job satisfaction of software engineers working in agile projects must have special attention to understand how these affect the achievement of the project's goal (Cao and Park, 2017; Serebrenik, 2017; Tripp et al., 2016). Besides, when agile software engineers are aware of their affective states, causes, and consequences at work, they could take actions to regulate those affective states to maintain a desirable level of performance, improve communication and teamwork, or increase the software quality (Pedrycz et al., 2011; Crawford et al., 2014; Guzman and Bruegge, 2013; Bishop and Deokar, 2014).

Literature in the field of psychology demonstrates that affective states (e.g., emotions, moods, and feelings) influence individuals' cognitive processes and behaviors (e.g., decision-making, problem understanding, and teamwork) (Ekman and Davidson, 1994; Rodríguez et al., 2011). Regarding the Software Engineering (SE) literature, there are works that have studied the influence of affective states on software engineers' performance using mainly traditional methodologies (e.g., waterfall, spiral or incremental) (Graziotin et al., 2018; Druskat and Wolff, 1999). In this context, investigating and understating the role of human aspects such as affective states, their influence, and consequences on performance, and the success or failure of agile projects become of paramount relevance given the increasing adoption of the ASD paradigm by the software industry (Graziotin et al., 2014b; Cockburn and Highsmith, 2001; Mulder, 2020; Alhubaishy and Benedicenti, 2017b). Therefore, we considered that the classification and organization of the recent literature on the study of software engineers' affective states on agile environments is crucial to identify the agile software development project activities influenced by these affective states and their consequences.

In this article, we present a Systematic Mapping Review (SMR) to map out and understand the existing research about affective states and their influence on agile software engineers' performance and the quality of the developed software. We adopted a *mapping review* approach by addressing the four elements of the PICO strategy (Kitchenham, 2004; Petersen et al., 2008) to formulate our main research question and achieve the objective of this SMR: *What affective states influence agile software engineers' performance and developed software quality?* To answer this question, we analyzed studies published between January 2010 and September 2021. In particular, we retrieved over 16,000 articles and after applying selection criteria, we selected and analyzed 24 primary articles. Furthermore, a classification scheme was designed based on the information extracted to identify, characterize, and summarize the empirical evidence that addresses our main research question.

The remainder of this article is structured as follows. In Section 2 we discuss related work. In Section 3 we present the research method utilized to conduct this SMR. We answer the main and specific research questions in Section 4 and discuss relevant findings in Section 5. Finally, in Section 6 we provide some concluding remarks and future work.

2. Related work

The literature reports some systematic reviews investigating the role of affective states (e.g., emotions) in software engineering (see Table 1). Sánchez-Gordón and Colomo-Palacios (2019) conducted a SLR to investigate the role of emotions in software engineers to (i) understand the emotional facet of software developers, (ii) find out the most reported methods for emotion measurement, and (iii) determine whether there is a growing interest in the field. Results indicated that (i) the most frequent emotions experienced by software engineers are anger, fear, disgust, sadness, joy, love, and happiness; (ii) most of the studies used software practitioners and data sets from industrial context as subjects; and (iii) there is not agreed approach to measuring emotions in SE.

There are systematic reviews that focus on investigating other human factors besides emotions. Lenberg et al. (2015) focused on software engineers' cognitive, social, and behavioral aspects (e.g., motivation, stress, and personality). Lenberg et al. (2015) found that the existing behavioral research can help create a

Table 1
Related work (For Lenberg et al. (2015), we list 21 from their 55 major terms).

Ref.	Years	Type	Journal/ Conference	Topics	PICO	Research questions	Search string (major terms topics)	Reviewed articles	Primary studies
Sánchez-Gordón and Colomo-Palacios (2019)	2005–2018	SLR	Journal	- Emotions. - Software engineering.	NO	RQ1. What is the trend of studies related to developers' emotions that have been reported in major bibliographic databases? RQ2. What type of research methods are used in the studies? RQ3. What is the citation landscape of the studies in this area? RQ4. What are the software developers' emotions addressed or investigated that have been reported in the studies? RQ5. How software developers' emotions are measured?	- Emotions. - Software engineering.	7172	66
Lenberg et al. (2015)	1997–2013	SLR	Journal	Behavioral software engineering.	NO	N/A	- Cognitive style & affective - Job insecurity & satisfaction - Life satisfaction - Organization - Self control, discipline, efficacy, esteem, monitoring - Personality - Work life balance - Workplace happiness - Cohesion - Communication & Conflicts - Goal orientation theory - Motivation & Stress - Social value orientation	Over 10 000	250
Hidel-laarachchi et al. (2021)	1997–2019	SLR	Journal	- Human aspects. - Software requirement process.	YES	RQ1. What is the motivation behind each primary study on identifying the effects of human aspects in RE? RQ2. What is the current status of research studies on the effect of human aspects in RE? RQ3. What RE phases are most impacted by human aspects, and what are the relationships between different human aspects that affect these RE phases? RQ4. How do the identified human aspects affect the RE Process?	- Human aspects. - Human factors in requirements engineering - Effects on human aspects on RE	472	74
Kuutila et al. (2017)	?? –2016	ILR	Conference	- Time pressure. - Job satisfaction. - Workplace. - Software engineering.	NO	N/A	- Time pressure.	1378	1270
Hall et al. (2009)	1980–2006	SLR	Journal	Motivation in software engineering.	YES	RQ1: What are the characteristics of software engineers? RQ2: What (de)motivates software engineers to be more (less) productive? RQ3: What are the external signs or outcomes of (de) motivated software engineers? RQ4: What aspects of Software Engineering (de) motivate software engineers? RQ5: What models of motivation exist in Software Engineering?	- Software engineering. - Personality. - Motivation. - Productivity measures.	Over 2000	92

(continued on next page)

Table 1 (*continued*).

Ref.	Years	Type	Journal/ Conference	Topics	PICO	Research questions	Search string (major terms topics)	Reviewed articles	Primary studies
Soomro et al. (2016)	1993–2015	SLR	Journal	- Personality traits. - Software team climate. - Software team performance.	YES	RQ1. What, if any, are the key software engineer personality trait(s) that have a significant influence on software team climate and/or software team performance? RQ2. What are the key factors in team climate composition that have been addressed or investigated in software development team studies? RQ3. Which, if any, software team climate factor(s) has/have significant effect on software team's performance? RQ4. How has software team's performance been measured in these studies?	- Personality. - Software engineering. - Software teams. - Software team climate.	Over 10 000	35

richer understanding of how software engineers behave, think, and feel. Finally, they suggest that researchers should explore more human aspects and consider their influence on a broader range of SE activities. Similarly, Hidellaarachchi et al. (2021) focused on the effects of human aspects and their consequences in Requirements Engineering (RE) processes. Their findings revealed that the human aspects that are most taken into account in RE are communication issues, personality, human values, gender, motivation, emotions, and culture.

Furthermore, they found that the lack of motivation and domain knowledge, the heavy task workload, human errors, and restrictions on knowledge sharing are other human aspects that can negatively affect RE. For example, negative emotions affect user rejection, reliability, and requirements stability.

Hall et al. (2009) conducted a SLR related to human aspects in SE to identify the theory used in studies investigating the motivations of software engineers to help understand how to maximize software engineers' performance for the benefit of the organizations. Hall et al. (2009) concluded that most motivational theories are not used in SE and the understanding of motivation in SE remains immature, and without this understanding, it is hard to adapt their application to software engineers. In addition, Soomro et al. (2016) conducted an SLR to investigate another human aspect of SE: software engineers' personalities. Their results showed a relationship between software engineers' personalities and team performance. For example, introverted team members and their lack of interpersonal communication skills decrease the effectiveness of software development, but extroverted team members can build a successful software development team because of their communication, confidence, and compromise skills.

As shown in Table 1, most of these studies have focused on various human aspects related to SE, such as emotions (Sánchez-Gordón and Colomo-Palacios, 2019; Lenberg et al., 2015; Hidellaarachchi et al., 2021), behavior (Lenberg et al., 2015), motivation (Hall et al., 2009), personality (Lenberg et al., 2015; Soomro et al., 2016); however, there are also systematic reviews investigating how external phenomena (e.g., time pressure Kuutila et al., 2017) influence engineers' affective states. For example, Kuutila et al. (2017) conducted an Interdisciplinary Literature Review (ILR) to investigate the influence of time pressure in SE and other fields. The results in Kuutila et al. (2017) indicated that time pressure has a negative effect on job satisfaction, self-reported health, burnout, and the emotional well-being of software engineers and their performance and goal fulfillment.

From the previous studies, we noted that the authors did not consider (i) if there is a distinction between the software engineers' human factors or affective states when they work in different software development environments (e.g., agile or traditional) or (ii) how affective states or human factors influence the software quality, except for Soomro et al. (2016). On the other hand, the authors found that (i) most of their primary studies used practitioners or academic datasets to collect data instead of industry datasets, (ii) human factors impact software engineers' performance in the software development processes or decision-making, and (iii) the physiological measures or behavioral measures are the least reported tools in SE.

As shown, very few related SLR focus on the influence of affective states in agile environments. Most of them are focused on human factors that can influence software engineers' affective states and affect software projects in diverse SE contexts since these have a relationship with human affective states, such as personality (Graziotin et al., 2014b; Alhubaishy and Benedicenti, 2017b), and motivation (Sharp et al., 2009; Beecham et al., 2008). Furthermore, the relationship between human aspects (e.g., emotions, personalities, behavior, and motivation) and software engineers' performance is currently under research. Particularly, our SMR represents an effort to show the relationship between affective states and their consequences on agile software engineers' performance and software quality.

3. Research method

The present SMR follows the guidelines proposed by Petersen et al. (2008). This study focused on the influence of agile software engineers' affective states on their performance and the quality of developed software. We follow the PICO strategy (Kitchenham, 2004) to frame our research questions.

3.1. Definition of research questions

The main research question of this SMR is the following: *What affective states influence agile software engineers' performance and developed software quality?* We formulated the following research questions to provide a more specific understanding of the topic investigated.

- **RQ1.** *When were the primary studies published?*
- **RQ2.** *What affective states are considered in primary studies?*
  - **RQ2.1** *What agile methodologies are related to affective states in primary studies?*
  - **RQ2.2** *How is the agile software engineers' affective data collected?*

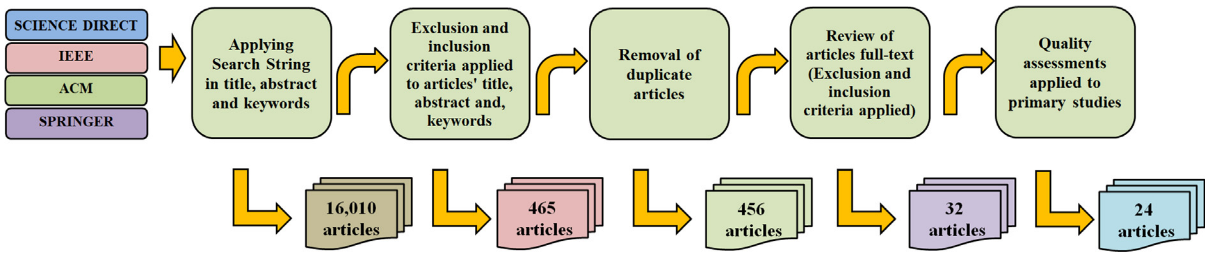


Fig. 1. Search strategy process.

- **RQ3.** *What agile project activities and development activities are influenced by agile software engineers' affective states?*
- **RQ4.** *What are the software engineers' performance and software quality metrics related to affective states in agile environments?*
- **RQ5.** *What are the limitations of primary studies to study affective states in agile environments?*

3.2. Search strategy

The search strategy of the primary studies included (i) the definition of the major search terms, (ii) the search string construction, and (iii) the selection of digital scientific databases. We defined a search string according to the guidelines proposed by Kitchenham (2004), which suggest the PICO criteria to build a search string: *Population* (targeted group), *Intervention* (e.g., methodology, tools, technology, or procedure), *Comparison* (comparison between different interventions), and *Outcome* (e.g., scope, techniques, and security applications of studies analyzed). According to Kitchenham and Charters (2007), comparison and outcome could be omitted for a SMR since generally has broader research questions driving the SMR and often asks multiple research questions. Thus, we define the *Population* as agile software engineers; and the *Intervention* as the empirical studies involving the agile software engineers' affective states.

The search string was built as follows. First, we derived the major terms from the research questions. Second, we identified alternative spellings, synonyms, and related terms for major terms to join them using the Boolean OR operator. Finally, we used the Boolean AND operator to link the major terms. In addition, we decided to use the terms below since, through a preliminary literature review of agile methodologies publications, authors use terms such as "Scrum", "Kanban", "XP", or "agile frameworks" to refer to an agile methodology instead of using "agile", or "agile methodology". Regarding the rest of the search string (software engineers' affective states, software quality, and software engineers' performance), it was decided based also on the preliminary literature review.

We defined the search string as a combination of four sets of keywords focusing on (i) *software engineers affective states*, (ii) *agile environments*, (iii) *software engineer performance in agile environments*, and (iv) *quality metrics to evaluate a software*. The final search string is as follows:

- *Agile software methodologies:* ("agile methodology" OR scrum OR "extreme programming" OR "kanban" OR "dynamic systems development method" OR "agile approach" OR "agile framework" OR "agile method").
- *Software engineers' affective states:* (("facial expression recognition" OR "text based emotion recognition" OR "voice emotion recognition" OR "affective computing" OR "emotional data collection" OR "behavioral software engineering" OR "multimodal emotion recognition algorithms") AND ("software developer" OR "software engineer" OR "software engineering")).
- *Software engineers' performance:* ("software engineer efficiency" OR "sprint burndown" OR "team velocity metric" OR "team throughput" OR "software development cycle time" OR "lines of code" OR "developer's performance").
- *Software quality:* (("software functional suitability" OR "software reliability" OR "software operability" OR "software performance efficiency" OR "software security" OR "software compatibility" OR "software maintainability" OR "software transferability") AND metric).

Regarding the scientific databases, we selected ScienceDirect, IEEE Xplore, ACM, and Springer. It is worth mentioning that some search engines in digital libraries have limitations by using boolean operators when using complex search strings in their search fields. Therefore, we manually split the search string to overcome the search engines' limitations, considering that this action aimed to obtain the same results as the original search string. We include articles published from January 2010 to September 2021 since, through a quick review of agile methodologies publications, we found that since 2010 there have been more related publications.

3.3. Screening of articles for inclusion and exclusion

The inclusion criteria were defined as follows.

- Studies that considered software engineers' affective states (e.g., emotions, moods, or feelings).
- Studies where software engineers' affective states influenced the software engineers' performance or the software quality in agile projects.
- Studies related to the field of Software Engineering in agile environments.

The exclusion criteria were the following:

- Articles no published in journals and conference proceedings.
- Articles not available in English.

The selection of the primary studies (see Fig. 1) was an iterative process that involved the three authors. In the first step, the first author performed an automatic search by applying the search string in the fields of title, abstract, and keywords of the selected digital databases.

Secondly, to discard studies considered non-relevant for the mapping review, the first author applied the inclusion criteria to the title of the articles retrieved. We decided to perform the initial filtering by the article's title since the Springer search database did not apply the search string on a particular article's field; thus, the Springer search results included articles not related to the research area of this study. However, the article's title allowed us to determine its research area. Furthermore, if the first author was not wholly convinced, the inclusion criteria were applied to the abstract and the keywords. This step produced a set of articles that we considered candidates for inclusion as primary studies. Once the first author included the relevant studies, the remaining authors used the exact search string in the selected digital libraries to verify that the initial number of obtained articles was the same. The authors verified the correct application of inclusion and exclusion criteria.

In a third step, we imported the citations obtained into Mendeley 1.19.8.0[1] to remove all the duplicate copies automatically. Finally, as a fourth step, the first author identified and selected primary studies by applying the inclusion criteria again. However, this time, the first author read the full text of the selected studies to confirm if the selected articles have relevant information for the study. The second and third authors reviewed each article to reduce the risk of incorrectly including or excluding an article. In those cases where these two authors had conflicting opinions, a third author reviewed the article and made a final decision. This step resulted in a final set of articles considered primary studies in Section 4.1.

3.4. Quality assessment

Regarding the quality assessment, we contrasted the terms referring to emotions, feelings, and moods (e.g., "affect", "feeling", "emotions", or "sentiments") with those presented in Ekman's six emotions framework (Ekman, 1992), Plutchik's wheel of emotions (Plutchik, 1991, 2001), Russell's circumplex model of affect (Russell, 1980), and the Parrot's emotion classification framework (Parrott, 2001) to ensure that these terms refer to affective states, thus avoiding confusion between their meanings. However, some terms could not be specific enough to distinguish whether they are affective states since not all the authors use an affective framework (we use this term for convenience to refer to the previous frameworks) to define the affective states they reported.

3.5. Keywording

Petersen et al. (2008) defined the keywording process as a way to reduce the time needed in developing a classification scheme and ensuring that such a scheme considers the existing studies. We did keywording in two iterative steps. In the first step, after we selected the primary studies (see Section 4.1), we read the article's full-text and obtained keywords by finding common characteristics between these elements, which are related to the research questions (see Section 3.1). While doing so, we put the keywords extracted from the primary studies into a spreadsheet to develop a high-level understanding of the contribution of each research. Then, we classified each primary study into a same column when a new keyword coincided with another one in the spreadsheet. We repeated this process because, in the first revision of the primary studies, some of the previously identified keywords might not have been identified in the first ones since there was no awareness of their existence yet. Finally, we chose the keywords set and clustered it to form the categories for the map.

Afterwards, we designed a classification scheme based on the process of creating affinity diagrams (Plain, 2007) and on the data extracted from primary studies (see Section 4.1). The procedure was as follows. First, we put the keywords extracted from primary studies into a spreadsheet. Each keyword was recorded in an individual column and each primary study in an individual row. This allowed us to carry out the keywording process described above. These steps correspond to the steps of creating affinity diagrams (1) *Transfer data to index cards or sticky notes*, and (2) *Scatter the cards on a table or post the notes on a wall or board*.

Second, each author participated in defining the classification groups (see Table 7) that describe each of the grouped keywords and categorizing them according to affinity and research questions (see Section 3.1). In addition, each author performed periodic revisions of the keywords selection, category creation, and scheme design. These steps combine the step *(3) Divide the group of people into smaller groups of two or three*, step (4) *Have one group arrange the cards silently*, and step (5) *Rotate the group of participants after two or three minutes until the notes are organized into groups of related ideas, issues, or topics*, of creating affinity diagrams.

Finally, all the authors discussed the grouping given to each keyword and its relationship with the research questions (see Section 3.1) and participated in the design of the classification scheme based on the keywords' affinity and their categories. This step corresponds to the step of creating affinity diagrams (6) *Develop titles for each group of cards. If logical, put related groups into larger supergroups*.

It is worth mentioning that some steps of the process of creating affinity diagrams (as presented in Plain (2007)) were adapted when creating our proposed classification scheme. As an illustration, instead of *transferring data to index cards or sticky notes* (step 1) or *scattering the cards on a table* (step 2), we utilized a spreadsheet to record, organize, order, and scatter the keywords extracted from primary studies. Also, instead of *dividing the group of people (i.e., the four authors) into smaller groups* (step 3), we carried out the corresponding activities in several iterations in which all the authors were alternately participating.

3.6. Data extraction and mapping of studies

The classification scheme evolves while doing the keyword extraction, adding new categories or merging and splitting the existing ones (into a spreadsheet). From the final spreadsheet, we analyzed the results and obtained the publications frequency per category to see the ones emphasized in past research and answer the research questions. These categories allowed us to identify possible gaps and future research topics.

4. Results

In this Section, we present the results of the systematic mapping organized in the following subsections: 4.1 Summary of the results, 4.2 Answers to the research questions, and 4.3 Classification scheme of influence of affective states in agile software engineers' performance and software quality.

4.1. Summary of the results

4.1.1. Search results

Table 2
Search strategy process.

Database	Science direct	IEEE	ACM	Springer	Total
Automatic search	654	2341	1471	11,544	16,010
Inclusion & exclusion criteria (title, abstract, and keywords)	139	84	46	196	465
Removal of duplicates	139	84	40	193	456
Inclusion & exclusion criteria (full-text article)	11	9	4	8	32
Quality assessments	8	7	4	5	24

[1] Mendeley 1.19.8.0: https://www.mendeley.com/autoupdates/installers/preview.

We retrieved an initial set of 16,010 articles from the automatic search step. After applying the inclusion and exclusion criteria to the article's title, abstract and keywords, a total of 465 articles were selected. Subsequently, we removed nine duplicated studies using Mendeley (456 articles remain). Then, we applied the inclusion and exclusion criteria to the full-text articles to discard non-relevant studies, remaining a total of 32 articles. After that, we applied the quality assessment to these 32 primary studies, resulting in a set of 24 articles that represents the final set of primary studies analyzed in this mapping review. There were a total of 14 conference articles (58%) and 10 journal articles (42%). In addition, Table 2 and Fig. 1 present a summary of this process. Furthermore, Table 2 shows the summary of the 456 selected articles by digital library. In particular, 10% of the studies remained from the *ACM* digital database, 6% from the *Science direct* digital database, 8% from the *IEEE* digital database, and 3% *Springer* digital database.

4.1.2. Selected articles

We classified the 24 primary studies into Articles Type-A (58.3%) (see Table 3) and Articles Type-B (41.7%) (see Table 4) according to their main objective. Articles Type-A focused on studying the influence of affective states (e.g., emotions, feelings, or moods) on the agile software engineers' performance and the result of their projects. On the other hand, studies Type-B did not focus primarily on the study of affective states; however, the authors of this type of study provide evidence or conclude that affective states experienced by software engineers could be related to their performance or the software quality.

Table 3
Primary studies Type-A: Articles where authors studied the affective states alongside software engineers' performance or software quality.

No.	Title	Ref
P6	Positive affect through interactions in meetings: The role of proactive and supportive statements	Schneider et al. (2018)
P8	Do Agile Managed Information Systems Projects Fail Due to a Lack of Emotional Intelligence?	Luong et al. (2021)
P9	Unhappy developers: Bad for themselves, bad for process, and bad for software product	Graziotin et al. (2017c)
P10	Consequences of unhappiness while developing software	Graziotin et al. (2017a)
P11	Agile and Wellbeing-Stress, Empowerment, and Performance in Scrum and Kanban Teams	Laanti (2013)
P12	Toward a model of emotion influences on agile decision making	Alhubaishy and Benedicenti (2017a)
P13	Sensing developers' emotions: The design of a replicated experiment	Girardi et al. (2018)
P15	Teams that finish early accelerate faster: a pattern language for high performing scrum teams	Sutherland et al. (2014)
P16	Outcomes of emotional content from agile team forum posts	Marshall et al. (2016)
P17	Effective user stories are affective	Kamthan and Shahmir (2017)
P19	Stress in agile software development: practices and outcomes	Meier et al. (2018)
P20	On the unhappiness of software developers	Graziotin et al. (2017b)
P21	Chat activity is a better predictor than chat sentiment on software developers productivity	Kuutila et al. (2020)
P24	Individual differences limit predicting well-being and productivity using software repositories: a longitudinal industrial study	Kuutila et al. (2021)

4.2. Answers to the research questions

In this Section, we answer the research questions that lead this SMR (see Section 3.1).

Table 4
Primary studies Type-B: Articles where authors did not focus on the study of affective states, but they considered them as something related to the software engineers' performance or software quality.

No.	Title	Ref
P1	Successful extreme programming: Fidelity to the methodology or good teamworking?	Wood et al. (2013)
P2	Quality requirements challenges in the context of large-scale distributed agile: An empirical study	Alsaqaf et al. (2019)
P3	Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors	Drury-Grogan (2014)
P4	Investigating the relationship between personalities and agile team climate of software professionals in a telecom company	Vishnubhotla et al. (2020)
P5	Individual empowerment of agile and non-agile software developers in small teams	Tessem (2014)
P7	Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation	Mishra et al. (2012)
P14	Toward a model of emotional contagion influence on agile development for mission critical systems	Alhubaishy and Benedicenti (2017b)
P18	Challenges to teamwork: a multiple case study of two agile teams	Gulliksen Stray et al. (2011)
P22	Behavior-driven dynamics in agile development: The effect of fast feedback on teams	Kortum et al. (2019)
P23	Issues and challenges impacting the successful management of agile-hybrid projects: A grounded theory approach	Sithambaram et al. (2021)

RQ1. When were the primary studies published?

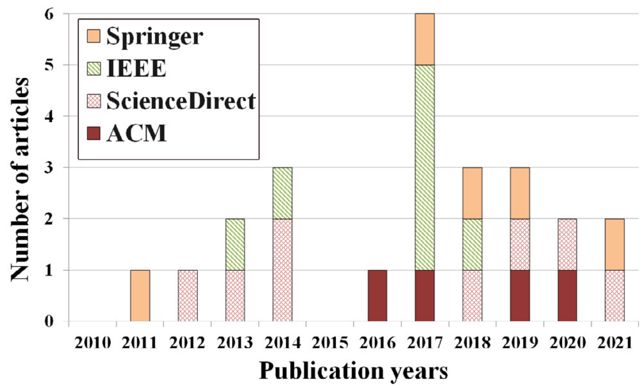
Fig. 2(a) shows the distribution of the 24 primary studies over the years by digital database. We observed that most primary studies (71%) are between 2016 and 2021. In addition, in 2017, there are more publications on the software engineers' affective states in agile environments in IEEE Xplore (16%). Similarly, Fig. 2(b) shows that since 2016, there are more studies Type-A (50%) than Type-B (21%).

RQ2. What affective states are considered in primary studies?

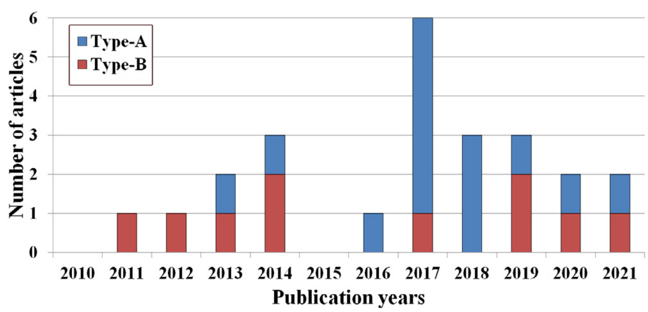
Table 5
Affective states in primary studies classified using Russell's circumplex model of affect (Russell, 1980).

Type		Affective states
Emotions	Positive	Joy [P21, P24]; Surprise [P21, P24] Happiness [P9, P10, P15, P20]; Happy [P1, P3]; Positive emotions [P6, P7, P12 P14, P16, P22, P23]
	Negative	Anguish [P17]; Confusion [P21, P24]; Sadness [P21, P24]; Anger [P21, P24], Anxiety [P4, P8, P10, P17]; Fear [P8, P10, P11, P23]; Frustration [P2, P5, P13] Unhappiness [P9, P10, P20]; Stress [P11, P13, P19]; Negative emotions [P6, P7, P12, P14, P16, P22, P23]
Feelings	Positive	Excitement [P1]
	Negative	Annoyance [P17]; Negative feeling [P18]
Moods	Positive & negative	Positive and negative moods [P4, P6, P22 P24]

In Table 5, we reported the affective states experienced by software engineers in agile environments (according to the literature). We classified the affective states as negative or positive based on Russell's circumplex model of affect (Russell, 1980). The 37.5% of primary studies did not use a specific emotion label (e.g., happiness or sadness), instead these indicated whether the affective state was positive or negative, so we decided to include them in the categories of positive or negative ones.



(a) Primary studies per digital library.



(b) Primary studies per type.

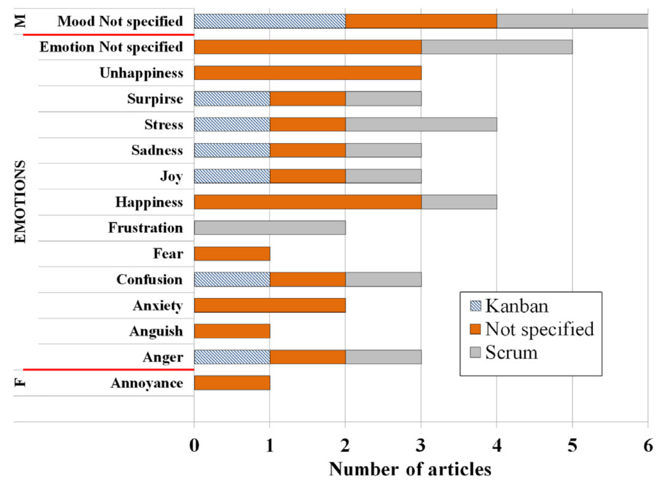
Fig. 2. Primary studies per year.

Table 5 shows that software engineers experience both negative and positive emotions, feelings, and moods. Therefore, the ability to effectively communicate and interpret emotions is central to daily human activities, and this ability is considered one of the key skills for functional team collaboration (Sutherland et al., 2014; Gulliksen Stray et al., 2011). For example, positive affective states might lead the agile software engineer to have good team collaboration to positively lead to enhanced leadership, improved communication, and good performance (Luong et al., 2021; Graziotin et al., 2017b; Sithambaram et al., 2021).

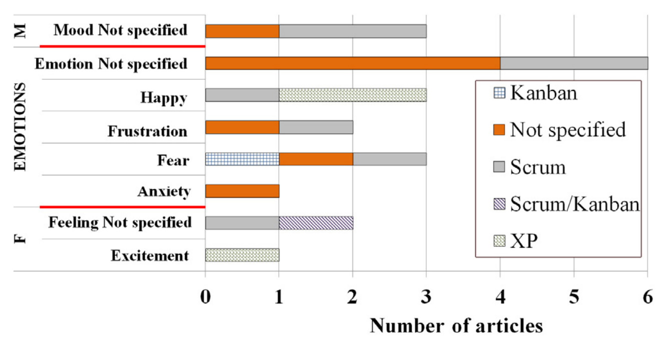
Moreover, the software engineers' happiness at belonging to a motivated group and the team satisfaction is evidence that the development team took full ownership of the development project, and everyone who worked on it was proud of what they did (Wood et al., 2013; Drury-Grogan, 2014; Tessem, 2014). On the contrary, when software engineers experience emotions such as frustration or stress, they could attempt to withdraw either temporarily or permanently from the project, or the company (Graziotin et al., 2017c,a,b).

A favorable work environment might reduce software engineers' stress levels by giving them good morale and the ability to manage the project's changing priorities and having low defect rates (Meier et al., 2018). In contrast, negative affective states disturb software engineers' performance and could lead them to delete their source code (Vishnubhotla et al., 2020; Graziotin et al., 2017b). Similarly, some agile software engineers experience fear or being afraid of their skills deficiencies which cause rejection for using some software tools, programming languages, software frameworks, or development methods used in projects (Luong et al., 2021; Graziotin et al., 2017b).

Fig. 3 shows that the affective states most mentioned in primary studies were emotions (80%). The rest of the articles focused on moods (14%) and feelings (6%). In addition, as we described in Table 5, we found more than one type of affective state in some primary studies. For example, in Graziotin et al. (2017a), we found anxiety and happiness, the same for Kuutila et al. (2020) where we found joy, sadness, confusion, and surprise. However, not all the articles mentioned a labeled affective state or its type; for example, in Alhubaishy and Benedicenti (2017a), Marshall et al. (2016), and Mishra et al. (2012), the authors studied positive and negative emotions without mentioning which emotions they were studying. Besides, Fig. 3 shows that emotions not specified (20%) and mood not specified (14%) were the affective states most mentioned in studies Type-A and Type-B. However, the authors did not label these affective states; they just specified them as positive or negative ones. The rest 66% (except for feeling not specified (3%)), correspond to labeled affective states, such as fear, joy, anxiety, surprise, or sadness.



(a) Primary studies Type-A



(b) Primary studies Type-B

Fig. 3. Affective states reported in primary studies per agile methodology. F = Feeling, M = Mood.

Moreover, Fig. 3(a) shows the affective states reported in studies Type-A per methodology, where the affective states most experienced by agile software engineers were not specified moods (14%), followed by not specified emotions (11%). The rest, 75%, were affective states such as happiness (9%), stress (9%), unhappiness (7%), surprise (7%), or anger (7%). In contrast to Fig. 3(a), Fig. 3(b) shows that the affective states most experienced by agile software engineers in studies Type-B were not specified emotions (36.4%), followed by fear (13.6%), happy (13.6%), and not specified moods (13.6%). The rest, 22.8%, corresponds to not specified feelings (9.1%), frustration (4.55%), excitement (4.55%), and anxiety (4.55%).

Furthermore, we found that not all the primary studies were based on an affective framework to describe or select the affective states. For example, eight (33%) of the primary studies mentioned the framework used, e.g., Plutchik's wheel of emotions (Kuutila et al., 2020, 2021) (8%), Russell's circumplex model of affect (Graziotin et al., 2017c,a; Girardi et al., 2018; Graziotin et al., 2017b) (17%), Xu and Wang's emotions (Alhubaishy and Benedicenti, 2017a) (4%), or the Parrot's emotion classification framework (Marshall et al., 2016) (4%) (see Fig. 4). The other 16 articles (67%) considered affective states without a framework as a reference; however, the affective states studied in these articles could be aligned with one of the above frameworks.

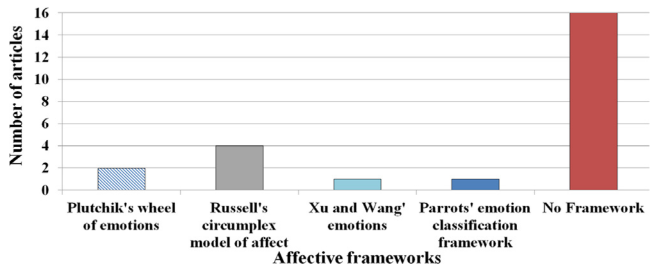


Fig. 4. Affective frameworks included in primary studies: Plutchik's wheel of emotions (Plutchik, 1991, 2001), Russell' circumplex model of affect (Russell, 1980), Xu and Wangs' emotions (Xu and Wang, 2007), and Parrot's emotion classification framework (Parrott, 2001).

RQ2.1 What agile methodologies are related to affective states in primary studies?

Fig. 5 shows the agile methodologies used in studies Type-A and Type-B. Fig. 5 shows that in most primary studies, authors specify the agile methodology used by software engineers (53.85%): Scrum (33.85%), Kanban (13.85%), XP (4.62%), and Scrum/Kanban (1.53%). The rest, 46.15%, corresponds to articles where the authors did not specify the agile methodology. In the studies, Type-A, 50% of the authors' articles did not specify the agile methodology they studied, while the rest of the studies, Type-A, 50%, correspond to Scrum (32%) and Kanban (18%). In contrast, from studies Type-B, the agile methodologies specified in them were Scrum (38%), XP (14%), Scrum/Kanban (5%), and Kanban (5%). The rest of the studies Type-B (38%) corresponds to not specified agile methodologies.

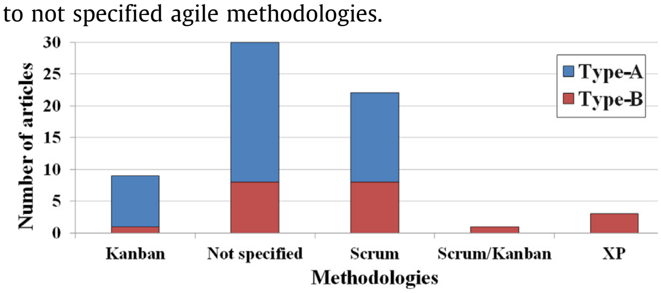


Fig. 5. Primary studies per agile methodology.

Fig. 3(a) shows that from the studies Type-A, where the authors did not specify the agile methodology they studied (50%), the 54.5% of software engineers' affective states experienced corresponded to negative affective states, 22.7% to positive ones, and the rest, 22.7% to unspecified affective states. Similar to the unspecified agile methodology (50%), in the studies where authors studied Scrum (31.8%), most of the affective states experienced by software engineers were negative ones (50%), and the rest corresponded to positive ones (21.4%) and unspecified ones (28.6%). On the other hand, in the studies corresponding to Kanban (18.2%), negative affective states continue to predominate (50%), and the rest of the affective states experienced by software engineers were positive ones (25%) and unspecified affective states (25%).

In contrast, studies Type-B (see Fig. 3(b)) where the authors studied Scrum (36.4%), 62.5% of software engineers' affective states experienced corresponded to unspecified negative affective states, 25% to negative ones, and the rest, 12.5% to positive ones. Similarly, where authors did not specify the agile methodology (40.9%), the 22.2% were negative affective states, and the rest, 77.8% unspecified. In studies Type-B where the authors studied XP (13.6%), 100% of affective states experienced by software engineers were positive. In contrast, in studies with Kanban methodology (4.5%), the affective states were negative (100%), and contrarily, studies with Scrum/Kanban (4.5%) had 100% of unspecified affective states.

RQ2.2 How is the agile software engineers' affective data collected?

Fig. 6 shows the different types of affective data gathering methods used in studies Type-A. In the 50% of these studies, the authors used methods such as questionnaires (Schneider et al., 2018; Alhubaishy and Benedicenti, 2017a; Kuutila et al., 2021; Sutherland et al., 2014), surveys (Luong et al., 2021; Graziotin et al., 2017c,a; Meier et al., 2018; Graziotin et al., 2017b), biometric sensors (Girardi et al., 2018), forum logs (Marshall et al., 2016), or chat messages (Kuutila et al., 2020). The rest 50%, did not specify the method used to collect affective data. We did not include the studies Type-B in Fig. 6 since they do not perform this activity of data collection.

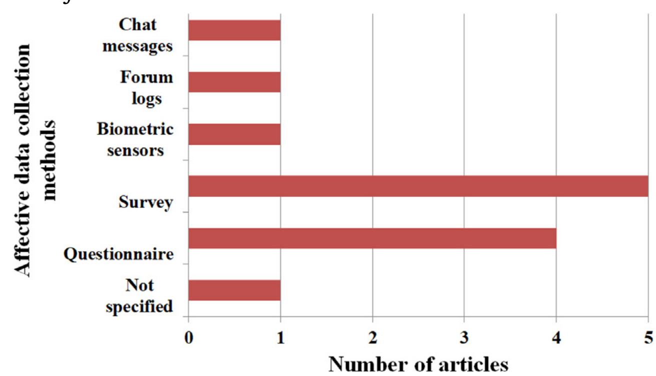


Fig. 6. Affective data collection methods in studies Type-A.

RQ3. What agile project activities and development activities are influenced by agile software engineers' affective states?

Results showed that agile software engineers' affective states influence agile software project activities (e.g., team composition or decision-making) and activities in the software development process (e.g., implementing user stories, coding, or fixing bugs) (Meier et al., 2018; Kuutila et al., 2020). We describe below the agile software development activities and the agile software project activities found in the primary studies.

4.2.1. Team composition

The design of a team requires considering human factors (e.g., motivation, knowledge, personality, behavior, and emotions) and their influence on the performance and the ability to work on agile teams (Meier et al., 2018; Gulliksen Stray et al., 2011; Kortum et al., 2019). According to Luong et al. (2021), affective states are fundamental in the software engineers' interaction with their teammates and how they work together. In addition, effective team performance depends on cohesion, adaptability, a good social atmosphere, good communication, and trust (product of managing their emotions and those of their peers to become more likely to be trusted) (Schneider et al., 2018; Vishnubhotla et al., 2020; Luong et al., 2021). Besides, when team members are aware of their own emotions, they can easily emphasize with their peers and provide support, consequently creating more team trust (Luong et al., 2021).

4.2.2. Decision-making

Negative emotions may influence executives or coworkers to make a wrong decision; for example, when they make an uninformed or ill-advised decision causing a "dirty" code-base change (Graziotin et al., 2017b). On the other hand, happy or unstressed agile teams can get a chance to participate in decision-making or involve themselves in higher-level decision-making that could significantly affect the process or the final product (Laanti, 2013; Wood et al., 2013; Tessem, 2014).

4.2.3. Effort estimation

Being perceived as incompetent for potentially making wrong estimates could make some agile software engineers' team members (e.g., junior team members) feel scared to make estimates or product backlog (Luong et al., 2021).

4.2.4. Implementing user stories

Agile software engineers who implement and improve their user stories could be less stressed since lower stress is linked to many software quality outcomes, such as low defect rate and good software architecture (Laanti, 2013; Sutherland et al., 2014; Kamthan and Shahmir, 2017; Meier et al., 2018). Besides, when software engineers' team members are happy, they would put "make better user stories" as the top priority improvement for the team (Sutherland et al., 2014).

4.2.5. Coding

Low code quality may result from software engineers' negative feelings and emotions, such as unhappiness or anger (Graziotin et al., 2017c).

4.2.6. Testing

Negative emotions such as frustration influence developers' performance when they are stuck in a debugging task (e.g., performance test or debugging test) or when they do not have the time to figure out why the code is not working (Alsaqaf et al., 2019; Alhubaishy and Benedicenti, 2017a; Graziotin et al., 2017b).

RQ4. What are the software engineers' performance and software quality metrics related to affective states in agile environments?

According to Graziotin et al. (2017c,a,b), Kuutila et al. (2020, 2021), Mishra et al. (2012) and Luong et al. (2021), affective states (e.g., emotions, feelings, and moods) and other human factors such as behavior and personality should be taken into account when the developer's performance is measured since they may fundamentally influence the agile team's ability to sustain a healthy working environment.

Table 6
Software engineers' performance & software quality in primary studies.

Category	Type A	Type B
Software engineers' performance	P6, P8, P9, P10, P11, P13, P15, P16, P17, P19, P20, P21, P24	P1, P3, P5, P7, P14, P18, P22, P23
Software quality	P6, P8, P9, P10, P12, P17 P19	P2, P4, P20, P23

Table 6 shows the primary studies where the authors considered the agile software engineer's performance (88%) organized into studies Type-A (62%) and Type-B (38%). In addition, Table 6 shows the primary studies where authors considered the quality of the developed software (46%). From these articles (46%), we list the studies Type-A (64%) and Type-B (36%).

Fig. 7 shows the metrics used in studies Type-A (see Table 6) to measure agile software engineers' performance (e.g., the number of commits and lines of code (Kuutila et al., 2020, 2021), the estimated or completed story points (Kamthan and Shahmir, 2017; Meier et al., 2018), and from self-perceived progress (Girardi et al., 2018; Laanti, 2013)), and to evaluate the software quality (e.g., completed story points (Meier et al., 2018), defect rates (Meier et al., 2018), or code quality (Graziotin et al., 2017c,a)). Given that studies Type-B do not consider how to measure the agile software engineers' performance or software quality along with software engineers' affective states, studies Type-B are not included in Fig. 7.

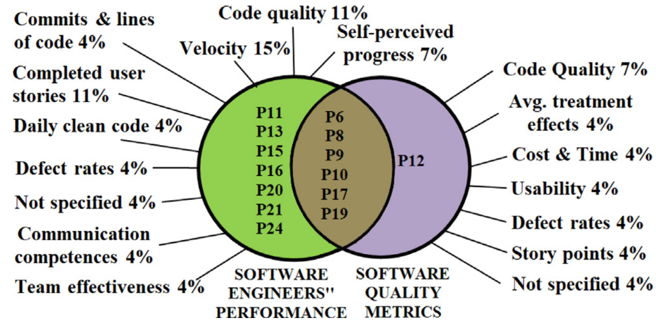


Fig. 7. Metrics used in studies Type-A to measure software engineers' performance and software quality. Percentages represent the number of articles where they used the metric.

Happiness and positive emotions are related to less time fixing problems (e.g., bugs) (Graziotin et al., 2017a). In the same sense, high job satisfaction and motivation are consequences of a low-stress level (Laanti, 2013). In contrast, poor cognitive performance could result from negative emotions (Graziotin et al., 2017a). Therefore, increasing developers' emotional awareness can improve performance, resilience to failures, and well-being (Girardi et al., 2018).

Software engineers' negative emotions (e.g., anger) may lead to low software quality since they could take shortcuts in their activities and cause low code quality (e.g., messy, sloppy, or buggy code) or even destroy (delete) their code or an entire project producing glitches in communication or delays in the project's delivery (Graziotin et al., 2017c,a). On the contrary, lower stress on software engineers caused by good morale, an engaged customer, or effective meetings might produce a low defect rate and good software architecture (Meier et al., 2018).

RQ5. What are the limitations of primary studies to study affective states in agile environments?

Many of the selected articles stated that their results depended on the authors' interpretations since they had to trust that participants freely replied the truth when they answered the interviews, questionnaires, and self-report instruments (Wood et al., 2013; Vishnubhotla et al., 2020; Graziotin et al., 2017a,c,b; Meier et al., 2018; Laanti, 2013). However, the authors considered the participants' tendency to respond inaccurately or falsely to questions whenever they are pre-informed about the objectives of a study (Vishnubhotla et al., 2020). Hence, some primary studies were based on qualitative and subjective analyses (e.g., Luong et al. (2021), Schneider et al. (2018), Graziotin et al. (2017a) and Tessem (2014)).

Subjective analysis can even provide a more comprehensive view of software engineers' (perceived) emotional abilities, as their full range of emotional experiences across different contexts in life (Luong et al., 2021). Nevertheless, the articles' authors did not propose a specific relationship between the affective states and some specific consequences on the agile software engineers' performance or the software quality. However, they assumed this relationship could be generalized (Luong et al., 2021; Schneider et al., 2018; Graziotin et al., 2017a; Tessem, 2014).

4.3. Classification scheme of influence of affective states in agile software engineers' performance and software quality

In this Section, we present our classification scheme (see Fig. 11) based on the analysis of the primary studies (see Section 3.5) and on the answers to the research questions (see Section 4.2).

Table 7
Significant categories and keywords.

Software quality	Software metrics	Code quality Functionality Testability Effectiveness Maintainability Reliability Portability & Compatibility Usability Story points Complexity Time Cost Others
Agile software engineers' performance	Team	Organization Orientation Communication Cohesion Diversity Roles Feedback Collaboration
	Individual	Performance Decision-making Responsibility Knowledge Leadership Motivation Behavior Personality
Affective states	Emotions, feelings and moods	Moods or feelings Negative emotions Positive emotions Affective states Undefined Team satisfaction
	Consequences in software development	Influences in cognitive processes Mental unease or disorder influences in motivation Influences in performance Communication challenges Low code quality Consequences undefined

4.3.1. Significant keywords

We obtained an initial set of 245 keywords from the analysis of the 24 primary studies (see Section 3.5). Our criterion for grouping the important keywords was finding common characteristics between these elements. After grouping them by affinity, we obtained 41 significant keywords (see Table 7). For example, we grouped keywords such as culture, team composition, and team size as *Team diversity*.

4.3.2. Significant categories

We defined five classification groups (see Table 7) that describe each of the grouped significant keywords and categorized significant keywords according to affinity and research questions. For example, we grouped significant keywords such as code quality, testability, and usability in the *Software quality category*; and keywords such as moods, feelings, mental unease, and disorder in the *affective states category*. We defined three categories from the initial five categories:

- *Affective states*. This category groups what the software engineers experienced or felt about doing activities and the consequences (e.g., influences in cognitive processes, mental unease or disorder, or low-quality code).
- *Agile software engineers' performance*. This group includes the skills related to the software engineers' performance, such as leadership, teamwork, communication, decision-making, collaboration, or problem-solving.
- *Software quality*. This category groups the measures used to evaluate software characteristics (e.g., code quality, testability, usability, or reliability) affected by agile software engineers' affective states.

Subsequently, based on the 41 significant keywords, the five classification categories, and the affinity diagram (Plain, 2007), we analyzed the primary studies to establish the relationships between the significant keywords and the type of influence between them (i.e., positive, negative, or both). We discarded some keywords and categories for the design of our classification scheme (see Fig. 11), such as the methods to collect or analyze affective data, since regardless of how the authors measured or identified the software engineers' affective states, this is not relevant to know how they influence the agile software engineers' performance or the software quality.

4.3.3. Description of the classification scheme

Based on the significant keywords and categories (*affective states, agile software engineers' performance, and software metrics*), we designed the classification scheme of the influence of affective states on agile software engineers' performance and software quality (see Fig. 11) to answer the research questions (see Section 3.1). In this sense, we organized each significant keyword into a significant category, starting with the most general category and breaking it down into specific subsets. Then, we discussed the grouping given to each significant keyword and its relationship with the research questions. The classification scheme is divided into two parts: (i) the first describes the influence of software engineers' affective states on their performance, and (ii) the second describes the influence on software quality.

Affective states. Affective states such as feelings are important because, whatever software engineers perceive, their affective states are related to their experiences and environment (e.g., in the company) (Laanti, 2013). For example, when software engineers share what would make them feel happy, the team could estimate the value of those things and search strategies for doing it, such as minimizing work in progress by focusing just on the completion of one user story or making them feel more confident in what they are doing (Sutherland et al., 2014).

Software engineers can use their emotions to drive their behaviors to achieve their goals (Luong et al., 2021). For example, when software engineers focus their emotions on solving a complex problem when none of their teammates can do it, the software engineers could increase their team knowledge and reduce their stress levels by showing them relevant information that they could use in future decisions (e.g., in having control of their work) (Laanti, 2013; Tessem, 2014; Mishra et al., 2012).

Performance. In Fig. 11, we describe how the agile software engineers' performance is influenced by their affective states. Whenever software engineers experience an affective state, whether positive or negative, their performance will be affected (Kuutila et al., 2020).

Affective states influence the cognitive processes of software engineers (see Fig. 8), such as problem understanding, decision-making, team collaboration, or lack of cohesion (Alhubaishy and Benedicenti, 2017a). These processes are associated with software engineers' performance, and software quality (Graziotin et al., 2017c,a; Laanti, 2013; Meier et al., 2018). For example, low cognitive performance is a consequence of negative emotions (e.g., frustration) because they may lead to unclear thinking of software engineers while they get more mentally fatigued (Graziotin et al., 2017c,a,b). In addition, negative feelings lead to not thinking things clearly; for example, software developers' skills drop off as becoming more frustrated until, eventually, they close the Integrated Development Environment (IDE), where they are coding the project, and come back the next day to work on it (Graziotin et al., 2017c). Even anxiety is associated with priority management (e.g., bugs to correct, from minor to critical), but happiness and positive emotions are associated with less time at fixing issues (Graziotin et al., 2017a).

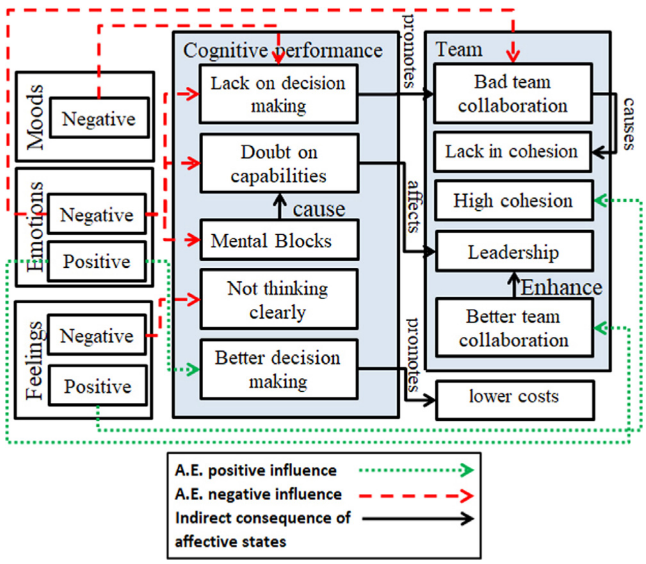


Fig. 8. Classification scheme of influence of affective states on cognitive performance & team's collaboration.

Positive emotions, culture, and personality can influence the decisions taken by agile teams at tactical, operational, and strategic levels, which could have essential effects on software outcomes, such as higher quality, lower cost, and shorter development time for successful software (Alhubaishy and Benedicenti, 2017a). Similarly, the lousy decision-making could make the software engineer feels unhappy (Graziotin et al., 2017b). On the other hand, the literature reported that developers enjoy working with challenging and diverse tasks, but if those tasks are irrelevant, developers may experience frustration (Graziotin et al., 2017b; Tessem, 2014). When developers experience emotions that lead them to unhappiness (e.g., frustration or stress), they could suffer a mental block and make them doubt their capabilities or attempt to withdraw either temporarily or permanently from the project or even from the company (Graziotin et al., 2017c,a,b). For example, Fig. 9 represents the association between job satisfaction and withdrawal, where low job satisfaction has implications for increasing employee turnover or withdrawal (Kuutila et al., 2021). On the other hand, low-stress levels produce high job satisfaction and motivation in software teams (Laanti, 2013). Team satisfaction is evident when the team is proud of what they did or when they are happy to say that they worked on a specific project (Wood et al., 2013; Drury-Grogan, 2014).

Interpersonal relationships are also an essential factor that affects the software engineers' performance, mainly because there is a great need to interact with the team members of a project (Vishnubhotla et al., 2020; Graziotin et al., 2017b). For example, some agile team members experience fear or high-stress levels caused by their skill deficiencies, which affect their ability to self-regulate their emotions and understand agile practices, such as daily stand-up meetings, onsite customers, or the use of storyboards that require direct and constant communication and collaboration (Schneider et al., 2018; Luong et al., 2021). Besides, software engineers' emotions could be affected by bad habits, dysfunctional communication behavior, or the team's overall atmosphere, which affects the quality and performance of software development teams (Kortum et al., 2019; Sithambaram et al., 2021). Moreover, a scrum master can benefit from understanding software engineers' affective states such as emotions by identifying and correcting uneven task distribution, supporting a team member in solving a task, or simply listening to their problems and proposing possible solutions (Girardi et al., 2018). Besides, some software engineers are sad or angry when they read others' code that they have to use, and they realize it is full of bugs or it does not compile anymore (Graziotin et al., 2017b).

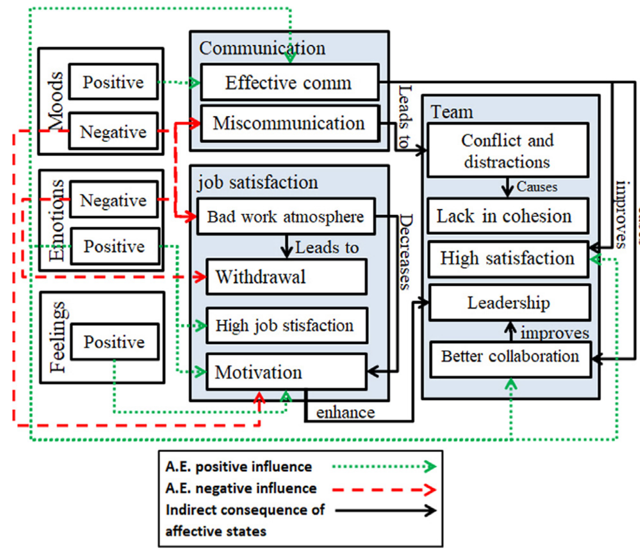


Fig. 9. Classification scheme of influence of affective states on communication, team collaboration and job satisfaction.

Software engineers' negative emotions like fear can be experienced in agile practice when software engineers require constant communication and collaboration (e.g., instant communication feedback, face-to-face communication or text-based communication) (Mishra et al., 2012; Luong et al., 2021). Fig. 9 shows that lack of cohesion is the consequence of conflicts and distractions caused by negative emotions of software engineers (Gulliksen Stray et al., 2011; Luong et al., 2021). Hence, to make agile software engineers happy to improve their collaboration, they must promote their team togetherness by improving their open communication and shared decision-making in agile development methods among team members (Sutherland et al., 2014; Mishra et al., 2012; Gulliksen Stray et al., 2011).

Software engineers' affective states, knowledge, diversity (e.g., culture, religion, or language), and context (e.g., physical environment) can positively influence the ability to work in teams (Mishra et al., 2012). In addition, software team performance would be less impacted when they could regulate their emotions (e.g., by motivating themselves) since the team could improve their communication and leadership effectiveness (see Fig. 8) (Sithambaram et al., 2021; Luong et al., 2021). Besides, software team members are more likely to be trusted and relied on for their competence, and ability (Luong et al., 2021). On the other hand, factors such as a lack of project goals and objectives, a weak team composition, a lack of team expertise or experience, negative emotions, or a lack of decision-making could lead the team to withdraw from the project (Drury-Grogan, 2014; Sithambaram et al., 2021).

Software quality. This part of the classification scheme (see Fig. 11) is about the metrics to evaluate software, metrics that could be affected by the software engineers' performance, and their affective states in agile environments (see some examples in Fig. 10). Finding the right balance between product quality and the software engineers' performance is an ongoing challenge in the agile approach (Bustard et al., 2013). For example, code quality (Meier et al., 2018; Graziotin et al., 2017a), complexity & correctness (Alsaqaf et al., 2019), testability (Wood et al., 2013), and usability (Kamthan and Shahmir, 2017). Besides, the biggest challenge of agile development is to ensure the software product quality and facilitate team effectiveness in each process (Drury-Grogan, 2014; Gulliksen Stray et al., 2011).

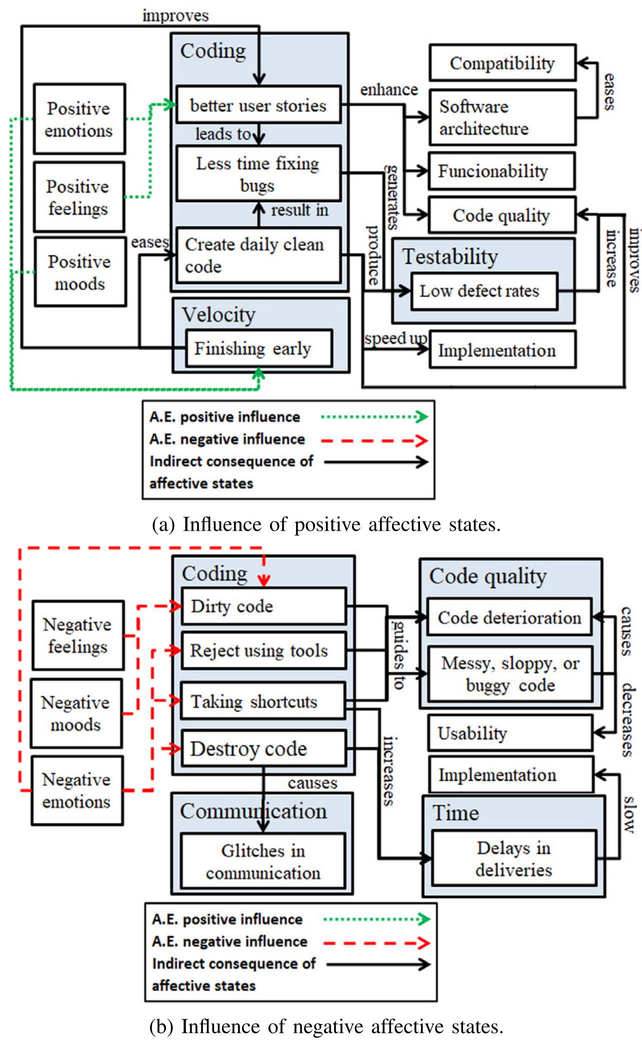
Affective states (e.g., emotions, feelings, or moods) may seem less relevant for software engineers at first glance; however, some selected studies (Schneider et al., 2018; Luong et al., 2021; Graziotin et al., 2017c; Meier et al., 2018) show that they are significantly related to several higher success rates much later in the project. For example, agile requirements engineering needs to consider instrumental quality attributes (such as usability or maintainability) along with software engineers' affective states and motivation (non-instrumental quality attributes) (Kamthan and Shahmir, 2017).

Graziotin et al. (2017c) showed that developers' negative emotions experienced while programming (see Fig. 10(b)), may be a source of several mental-related issues (e.g., low self-esteem, high anxiety, and burnout). Unhappiness could cause developers to take shortcuts in their activities, often leading to low code quality (e.g., messy, sloppy, or buggy code), which leads to software quality deterioration (Graziotin et al., 2017a). Moreover, some developers deleted their code and sometimes the entire software development project because they were angry, which may cause delays in the project or some glitches in communication (see Fig. 10) (Graziotin et al., 2017c). In contrast, a team will be less stressed if they create clean code daily and implement better user stories since they will waste less time fixing bugs (Sutherland et al., 2014; Kamthan and Shahmir, 2017; Meier et al., 2018) (see Fig. 10(a)). Besides, lower defect rates, good software architecture, and overall software quality are also associated with lower stress (Meier et al., 2018). Further, software quality could be affected by the low performance of software engineers, for example, when they are implementing incorrect or incomplete user stories because they are trying to implement as many as possible in a short time (Sutherland et al., 2014; Alsaqaf et al., 2019; Kamthan and Shahmir, 2017). However, if software engineers are happy, they can increase their velocity to finish early the assigned tasks (see Fig. 10) and improve the implementation of their user stories (Sutherland et al., 2014).

Fig. 11 shows the whole classification scheme, in which we represent and detail each of the essential elements found in the literature related to the influence of agile software engineers' affective states in their performance and the quality of the developed product.

5. Discussion

This study highlighted the influence of the affective state on software engineers' performance and the software quality in agile environments since these environments consider the software engineers' interactions among teammates to increase their performance (Beck et al., 2001; Uribe and Ayala, 2007). Similarly to the heart of agile (Cockburn, 2016; Alexander, 2016), we noticed some software engineers' human aspects besides affective states that could influence their individual or group performance during the realization of their tasks in agile environments, along with the quality of their results or software products (see Section 4.2): behavior, motivation and communication, and personalities (Vishnubhotla et al., 2020; Mishra et al., 2012; Luong et al., 2021). For example, agile software engineers working in teams experience and express different emotions during their tasks, which influence their performance since affective states allow individuals to adapt their actions to have better results in collaborative tasks (Graziotin et al., 2017c,a,b; Alhubaishy and Benedicenti, 2017a; Gulliksen Stray et al., 2011; Sithambaram et al., 2021).



(b) Influence of negative affective states.

Fig. 10. Classification scheme of influence of affective states on coding and software quality.

5.1. Agile software engineers' affective data collecting methods

Studies such as Fountaine and Sharif (2017), Salido Ortega et al. (2020) and Novielli and Serebrenik (2019) exposed the causes of lack of attention to the software engineers' emotions and their consequences in SE in general. Also, in Salido Ortega et al. (2020) is established the relationship between the software engineers' emotions and their performance within a software project. However, as we showed in Section 4.2, most of the methods used in primary studies to collect agile software engineers' affective data were surveys (Luong et al., 2021; Graziotin et al., 2017c,a; Meier et al., 2018; Graziotin et al., 2017b) or questionnaires (Schneider et al., 2018; Alhubaishy and Benedicenti, 2017a; Sutherland et al., 2014; Kuutila et al., 2021), which are considered intrusive since they could distract or reduce the agility of the software engineers in their tasks (Graziotin et al., 2017c; Zheng et al., 2019; Kołakowska et al., 2013). Consequently, we presume that these self-assessments only report what the software engineers feel or claim to experience when they answer these self-assessments. Besides, sometimes the agile software engineers only answer the self-assessments to fulfill the requirement without concern about what they answered and continue with their tasks (Girardi et al., 2018; Graziotin et al., 2017c,a,b).

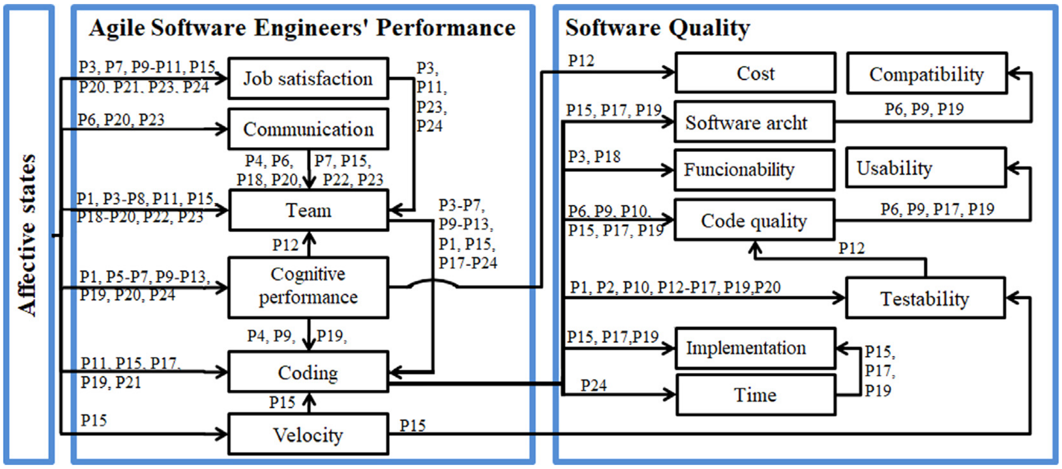


Fig. 11. Classification scheme of influence of affective states in agile software engineers' performance and software quality. The text around the arrows' scheme represents the papers on which the relations of the terms were based.

According to the agile manifesto (Beck et al., 2001; Uribe and Ayala, 2007), agile software engineers' priority is to satisfy the customer through early and continuous delivery of working software, even with changing requirements. So, we think agile software engineers must complete their activities (e.g., coding or fixing bugs) effectively and without distractions. This suggest that non-intrusive methods (e.g., biometric sensors (Girardi et al., 2018), commits and forum logs (Marshall et al., 2016), or chat messages (Kuutila et al., 2020)) should be used to collect the software engineers' affective data while developing their agile projects to not interfere with the software engineers' activities. For example, (i) using biometric sensors from devices such as the Empatica E4 wristband to measure the flow of electricity on the skin (Girardi et al., 2020), (ii) analyzing code commits from a GIT repository to identify the emotions that software engineers experienced while coding (Kuutila et al., 2020), or (iii) the team speed when counting completed story points over time (Hernández et al., 2019).

Moreover, we think that knowing the agile software engineers' affective states and contextual information while doing their tasks could contribute to knowing the causes of the software engineers' performance variation and provide them suggestions to regulate their affective states (Graziotin et al., 2017c,a,b; Luong et al., 2021). For example, agile teams can use the Niko-Niko calendar[2] to visualize how they are feeling on the week (by using smileys) to identify team members that are having problems with the project and help those team members, whether affectively or with the project's activities (Scott-Hill et al., 2020). Hence, a team leader or scrum master can make decisions about what to do when a team member experiences certain affective states or configure desirable contexts that allow for promoting desirable affective states (Ptaszynski et al., 2009; Schmidt et al., 1999).

5.2. Affective frameworks

In Section 3.4, we distinguished different types of affective states (e.g., emotions, feelings, and moods) by contrasting the terms referring to affective states (e.g., anger, surprise, joy, happiness) with those presented in Ekman's six basic emotions framework (Ekman, 1992), Plutchik's wheel of emotions (Plutchik, 1991, 2001), Russell's circumplex model of affect (Russell, 1980), and the Parrot's emotion classification framework (Parrott, 2001) to avoid confusion between the use of these terms. As a result, our evidence (see Section 4.2) showed that 16 primary studies (67%) considered affective states without an affective base framework. In the rest of the articles (33%), the authors mentioned the affective framework they considered in their studies, such as Russell's circumplex model of affect (Girardi et al., 2018; Graziotin et al., 2017c,a,b), Xu and Wang's emotions (Alhubaishy and Benedicenti, 2017a), or the Parrot's emotion classification framework (Marshall et al., 2016) (see Fig. 4). Therefore, when authors do not use an affective framework, they could confuse one type of affective state with another (e.g., moods to emotions or feelings) or use terms that seem real affective states (e.g., empowerment or uncomfortable).

Furthermore, we found that some authors studied a positive or negative group of affective states (e.g., Schneider et al. (2018)) without specifying how they classified those affective states in each group and without considering an affective base framework. This may lead to misclassifying an affective state and the group of affect (positive or negative) it belongs to. For example, the emotion "surprise" can be either positive or negative, depending on the context or the situation (Noordewier and Breugelmans, 2013; Ekman, 2004).

According to our results, there are articles where the authors used an affective framework to decide which affective states include in their studies (e.g., Graziotin et al. (2017c), Alhubaishy and Benedicenti (2017a), or Marshall et al. (2016)). Besides, some authors defined why they studied a specific affective state in their work. For example, in Sutherland et al. (2014), the authors measured happiness since they considered it one of the best predictive indicators of how people will behave in the future if the company is in trouble or doing something wrong. On the other hand, there are articles where the authors did not clarify why they used their selected affective states, such as Laanti (2013). Hence, we could presume the authors selected "stress" since it is a common affective state experienced at work or since software engineers are known to be stressed workers (Ostberg et al., 2020). Therefore, we consider that even if the authors define an affective framework or not, it is necessary to define why they are selecting an affective state or group of them.

5.3. Software engineers' performance and software quality in ASD

Both types of primary studies (see Tables 3 and 4) was useful to design our classification scheme (see Fig. 11) and highlight the influence of affective states on the agile software engineer performance and the quality of the developed software. For example, our classification scheme (see Fig. 11) shows that negative

[2] Niko Niko calendar: https://www.agilealliance.org/glossary/nikoniko/.

affective states (e.g., fear) could make the software engineers doubt their capabilities and make them reject using some software tools, programming languages, or development methods used in software projects (Graziotin et al., 2017b; Luong et al., 2021). Consequently, software engineers produce messy, sloppy, or buggy code, which affects the code quality or the testing phase (Graziotin et al., 2017c,a; Wood et al., 2013; Alhubaishy and Benedicenti, 2017b). On the contrary, positive affective states (e.g., happiness) could increase the software engineers' velocity and make them finish early their assigned task (see Fig. 10) to improve the implementation of user stories (Sutherland et al., 2014) which is related to less time fixing problems (e.g., bugs) and low defect rates (Graziotin et al., 2017a; Meier et al., 2018). Due to this, we presume that positive affective states influence software engineers' performance to produce better user stories which in turn produces software with good architecture and good functionality (Sutherland et al., 2014; Meier et al., 2018).

On the other hand, we found that when the agile software development processes do not contemplate engineers' affective states, it is possible that individual effectiveness will not be as good as several individuals working together in a team to achieve the same goals (Alhubaishy and Benedicenti, 2017a; Vishnubhotla et al., 2020). Besides, the inclusion of affective states in agile project management could benefit the agile software engineers and their team members by increasing their performance and software quality (Graziotin et al., 2017c,a,b). For example, a scrum master could identify and correct uneven task distribution or propose possible solutions to solving a task by supporting the team members by simply listening to their problems and understanding their emotions (Girardi et al., 2018). Also, when software team members can regulate their emotions (e.g., motivating themselves), the team will improve their communication and leadership effectiveness since they are more likely to be trusted and relied on for their competence and skills (Sithambaram et al., 2021; Luong et al., 2021).

According to our findings (see Section 4.2.6), affective states and human factors (e.g., behavior, personality, and motivation) must be taken into account when agile software engineers' performance is measured. However, just a few primary studies where authors studied affective states and agile software engineers' performance considered how to measure performance (e.g., the number of commits and lines of code (Kuutila et al., 2020, 2021), the estimated or completed story points (Kamthan and Shahmir, 2017; Meier et al., 2018), or self-perceived progress (Girardi et al., 2018; Laanti, 2013)) (see Fig. 7). Similarly, not all primary studies include metrics used to evaluate the quality of the developed software in agile projects, such as story points (Meier et al., 2018), defect rates (Meier et al., 2018), or code quality (Graziotin et al., 2017c,a) (see Fig. 7). Hence, we presume that agile software engineering research literature does not have well-defined metrics to measure the software engineers' performance and the software quality, or perhaps, authors' articles select or create their metrics depending on the team members or the kind of project they develop. Therefore, we considered that the metrics to measure performance or software quality should include the software engineers' affective states. As an illustration, affective states such as happiness and positive emotions are related to less time fixing problems (e.g., bugs) (Graziotin et al., 2017a) and negative emotions to low code quality or delays in the projects' deliveries (Graziotin et al., 2017c,a).

Identifying and regulating the software engineers' affective states in agile environments is a complex task that requires planning activities, such as redesigning tasks, flexible work schedules, or building cohesive work teams. Besides, software engineers must develop their self-awareness and training in individual and group skills and abilities to experience healthier affective states and minimize negative consequences.

5.4. Other human aspects

In addition to the affective states, we found other human factors that Software Engineering is currently considering as possible factors that may influence software engineers' performance in agile environments: behavior (Lenberg et al., 2015), motivation (Hall et al., 2009), personality (Soomro et al., 2016), time pressure (Kuutila et al., 2017), and its consequences on agile software engineers' activities within a software development project.

The first value and the third principle of the agile manifesto (Beck et al., 2001) recognize that building a cohesive team is more important than tools and processes to guarantee higher performance since working together empowers teams to take the imaginative leaps that produce innovative software. Hence, software engineers' positive affective states and human aspects (e.g., open communication and shared decision-making) are essential factors in agile environments that influence team collaboration, leadership, team cohesion, and the success or failure of agile projects (see Figs. 8 and 9) (Alhubaishy and Benedicenti, 2017a; Vishnubhotla et al., 2020; Mishra et al., 2012; Luong et al., 2021; Sutherland et al., 2014; Gulliksen Stray et al., 2011). In contrast, software engineers' negative affective states caused by time pressure in the project could affect their mental and emotional stability (Kuutila et al., 2017).

Moreover, the third value of the agile manifesto (Beck et al., 2001) gives higher importance to teamwork and directly engaging the customer in the work team. Similarly, our study highlighted that good communication facilitates software engineers to interpret the team members' emotions and properly collaborate with them and the customers during daily stand-up meetings (Luong et al., 2021). On the contrary, dysfunctional team behavior communication and low job satisfaction could cause bad communication (Mishra et al., 2012; Kortum et al., 2019), which affects the software development teams' performance (see Fig. 9) (Kortum et al., 2019; Sithambaram et al., 2021).

According to the twelfth principle of the agile manifesto (Beck et al., 2001), the work team is always willing to do changes to improve their effectiveness. Therefore, our study emphasized that it is necessary to consider the software engineers' affective states and human aspects since they can influence how software engineers do their activities or their results' quality (Graziotin et al., 2017c,a).

Furthermore, in Section 4.3, we showed evidence that communication could influence the software engineers' affective state and their motivation, or the team's overall atmosphere, which could influence the software development teams' performance and motivation during the fulfillment of different tasks such as coding, testing, or fixing bugs (see Fig. 10) (Laanti, 2013; Kortum et al., 2019; Sithambaram et al., 2021). Besides, when software team members could regulate their emotions and motivate themselves (e.g., improving communication and leadership effectiveness), team performance would be less impacted (Sithambaram et al., 2021; Luong et al., 2021). For example, whenever team members work and interact together, the ability to regulate their emotions influences establishing a collaborative environment where team members are encouraged to openly share and discuss their viewpoints or share knowledge and learn from each other (Luong et al., 2021).

In addition, we found evidence (see Section 4.3.3) showing that contextual information (information used to describe and characterize the agile software engineers' working environment) such as team-workers diversity (e.g., culture, or language), roles, team size, workplace (e.g., physical environment), or salary might influence agile software engineers' affective states (Wood et al., 2013; Drury-Grogan, 2014; Luong et al., 2021). Therefore, emotions, skills, and contextual information are key factors in designing agile teams (Luong et al., 2021). For example, when agile teams include software engineers with diverse expertise and roles (e.g., developers, software designers, architects, or testers), it is possible to understand the quality requirements challenges (e.g., miscommunication of the requirements or the lack of precise requirements) from different perspectives, and respond quickly to emergent situations (Wood et al., 2013; Alsaqaf et al., 2019).

Results (see Section 4.3.3) also show that software engineers' emotions, behaviors, judgments, and decisions could be explained by their personalities (e.g., openness to experience or conscientiousness) since personality allows software engineers to share their emotions with other team members, contributing to establish a good relationship between them when software engineers interact or solve problems together (Schneider et al., 2018; Drury-Grogan, 2014; Vishnubhotla et al., 2020).

6. Conclusions and future work

In this study, we conducted a Systematic Mapping Review and designed a classification scheme to represent and detail essential elements found in the literature related to the influence of the agile software engineers' affective states on their performance and software quality.

According to our research results, the agile software engineers' affective states influence agile software project activities, activities in the software development process, and the software outcomes, such as (i) leading the software engineers to stay or withdraw from the project or the company, (ii) increasing software quality or reducing the development time for successful software projects, (iii) causing fear and rejection of changes in development projects. Hence, we considered the study and identification of software engineers' affective states in agile environments crucial to achieving their well-being at work since agile principles are based on individuals (e.g., motivation, belonging, and trust) and their interactions over processes and tools. However, our results showed that agile software engineering research literature does not have well-defined metrics to measure the software engineers' performance and the software quality that considers the agile software engineers' affective states.

Our results also show that human aspects (e.g., behavior, communication, personality, and motivation) are important factors to note while evaluating software engineers' performance in agile environments since these human aspects can contribute to establishing good relationships between team members, sharing emotions, achieving goals, or even improving their performance.

Identifying agile software engineers' affective states could help the organizational managers (e.g., project managers or scrum masters) to develop strategies to determine the conditions in which software engineers experience certain affective states. In addition, the identification and regulation of affective states in the workplace could allow software engineers to understand, motivate and help their coworkers, promote their performance, learn from mistakes and manage work stress.

As future work, we plan to define the guidelines for analyzing the current metrics for software engineers' performance in agile environments and the methods and techniques for analyzing software engineers' affective states and performance in agile environments. In addition, we pretend to design and build a framework to determine the relationship between software engineers' performance, their affective states, and their possible causes by collecting software engineers' emotional and performance data in software development companies that use the agile methodology.

CRediT authorship contribution statement

Martín G. Salido O.: Conceptualization, Formal analysis, Investigation, Writing – original draft. **Gilberto Borrego:** Methodology, Writing – review & editing. **Ramón René Palacio Cinco:** Methodology, Writing – review & editing. **Luis-Felipe Rodríguez:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by the National Council of Science and Technology, Mexico (whose acronym in Spanish is CONACYT) of México, with scholarship number 786791 for the first author.

References

Alexander, C., 2016. Beyond the Agile Manifesto: The epoch of team. J. Def. Softw. Eng. 29 (6), 31–35.
Alhubaishy, A., Benedicenti, L., 2017a. Toward a model of emotion influences on agile decision making. In: 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 48–51.
Alhubaishy, A., Benedicenti, L., 2017b. Toward a model of emotional contagion influence on agile development for mission critical systems. In: 2017 International Conference on High Performance Computing & Simulation. HPCS, IEEE, pp. 541–544.
Alsaqaf, W., Daneva, M., Wieringa, R., 2019. Quality requirements challenges in the context of large-scale distributed agile: An empirical study. Inf. Softw. Technol. 110, 39–55.
Balijepally, V., Mahapatra, R., Nerur, S.P., 2006. Assessing personality profiles of software developers in agile development teams. Commun. Assoc. Inf. Syst. 18 (1), 4.
Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Ward, C., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., 2001. The agile manifesto. [Online]. Available: https://agilemanifesto.org/iso/es/manifesto.html.
Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H., 2008. Motivation in Software Engineering: A systematic literature review. Inf. Softw. Technol. 50 (9–10), 860–878.
Bishop, J., Cooper, K.M., Sharp, H., Whalen, M., 2020. Introduction to the special issue on software engineering in practice. Softw. Eng. Pract. 151–152.
Bishop, D., Deokar, A., 2014. Toward an understanding of preference for agile software development methods from a personality theory perspective. In: 2014 47th Hawaii International Conference on System Sciences. IEEE, pp. 4749–4758.
Bustard, D., Wilkie, G., Greer, D., 2013. Towards optimal software engineering: learning from agile practice. Innov. Syst. Softw. Eng. 9 (3), 191–200.
Cao, L., Park, E.H., 2017. Understanding goal-directed emotions in agile software development teams. In: Emotions in Agile Software Development Team.
Cockburn, A., 2016. Beyond the Agile Manifesto: The heart of agile. J. Def. Softw. Eng. 29 (6), 4–6.
Cockburn, A., Highsmith, J., 2001. Agile software development, the people factor. Computer 34 (11), 131–133.
Crawford, B., Soto, R., de la Barra, C.L., Crawford, K., Olguín, E., 2014. The influence of emotions on productivity in software engineering. In: International Conference on Human-Computer Interaction. Springer, pp. 307–310.
Diebold, P., Dahlem, M., 2014. Agile practices in practice: a mapping study. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. pp. 1–10.
digital.ai, 2021. 15th Annual State of Agile Report. Tech. Rep., CollabNet, Inc., [Online]. Available: https://info.digital.ai/rs/981-LQX-968/images/RE-SA-15th-Annual-State-Of-Agile-Report.pdf.
digital.ai (formerly CollabNet VersionOne), 2020. 14th Annual State of Agile Report. Tech. Rep., CollabNet, Inc., [Online]. Available: https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf.

Drury-Grogan, M.L., 2014. Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors. Inf. Softw. Technol. 56 (5), 506–515.
Druskat, V.U., Wolff, S.B., 1999. The link between emotions and team effectiveness: how teams engage members and build effective task processes. In: Academy of Management Proceedings, Vol. 1999. Academy of Management, Briarcliff Manor, NY 10510, pp. L1–L6, no. 1.
Ekman, P., 1992. Are there basic emotions? Psychol. Rev..
Ekman, P., 2004. Emotions revealed. Bmj 328 (Suppl S5).
Ekman, P.E., Davidson, R.J., 1994. The Nature of Emotion: Fundamental Questions. Oxford University Press.
Fagerholm, F., Münch, J., 2012. Developer experience: Concept and definition. In: 2012 International Conference on Software and System Process. ICSSP, IEEE, pp. 73–77.
Fountaine, A., Sharif, B., 2017. Emotional awareness in software development: Theory and measurement. In: 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 28–31.
Fowler, M., Highsmith, J., et al., 2001. The agile manifesto. Softw. Dev. 9 (8), 28–35.
Girardi, D., Lanubile, F., Novielli, N., Fucci, D., 2018. Sensing developers' emotions: The design of a replicated experiment. In: 2018 IEEE/ACM 3rd International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 51–54.
Girardi, D., Lanubile, F., Novielli, N., Quaranta, L., Serebrenik, A., 2019. Towards recognizing the emotions of developers using biometrics: the design of a field study. In: 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 13–16.
Girardi, D., Novielli, N., Fucci, D., Lanubile, F., 2020. Recognizing developers' emotions while programming. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. pp. 666–677.
Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2017a. Consequences of unhappiness while developing software. In: 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 42–47.
Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2017b. On the unhappiness of software developers. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. pp. 324–333.
Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2017c. 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). IEEE, pp. 362–364.
Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2018. What happens when software developers are (un) happy. J. Syst. Softw. 140, 32–47.
Graziotin, D., Wang, X., Abrahamsson, P., 2014a. Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ 2, e289.
Graziotin, D., Wang, X., Abrahamsson, P., 2014b. Software developers, moods, emotions, and performance. arXiv preprint arXiv:1405.4422.
Graziotin, D., Wang, X., Abrahamsson, P., 2015. Understanding the affect of developers: theoretical background and guidelines for psychoempirical software engineering. In: Proceedings of the 7th International Workshop on Social Software Engineering. pp. 25–32.
Gulliksen Stray, V., Moe, N.B., Dingsøyr, T., 2011. Challenges to teamwork: a multiple case study of two agile teams. In: International Conference on Agile Software Development. Springer, pp. 146–161.
Guzman, E., Bruegge, B., 2013. Towards emotional awareness in software development teams. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering. pp. 671–674.
Hall, T., Baddoo, N., Beecham, S., Robinson, H., Sharp, H., 2009. A systematic review of theory use in studies investigating the motivations of software engineers. ACM Trans. Softw. Eng. Methodol. (TOSEM) 18 (3), 1–29.
Hayat, F., Rehman, A.U., Arif, K.S., Wahab, K., Abbas, M., 2019. The influence of agile methodology (Scrum) on software project management. In: 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. SNPD, IEEE, pp. 145–149.
Hedberg Giffith, K., Nguyen, E., 2018. Sampling Affects of Software Developers to Understand Individual & Team Performance (Ph.D. dissertation). University of Gothenburg.
Hernández, G., Martínez, Á., Jiménez, R., Jiménez, F., 2019. Productivity metrics for an agile software development team: A systematic review. TecnoLógicas 22 (SPE), 63–81.
Hidellaarachchi, D., Grundy, J., Hoda, R., Madampe, K., 2021. The effects of human aspects on the requirements engineering process: A systematic literature review. IEEE Trans. Softw. Eng..
Kamthan, P., Shahmir, N., 2017. Effective user stories are affective. In: International Conference on Ubiquitous Computing and Ambient Intelligence. Springer, pp. 605–611.
Kitchenham, B., 2004. Procedures for Performing Systematic Reviews, Vol. 33. Keele University, Keele, UK, pp. 1–26, no. 2004.
Kitchenham, B., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering, Vol. 2. Tech. Rep., Keele University and Durham University Joint Report.
Kołakowska, A., Landowska, A., Szwoch, M., Szwoch, W., Wróbel, M.R., 2013. Emotion recognition and its application in software engineering. In: 2013 6th International Conference on Human System Interactions. HSI, IEEE, pp. 532–539.
Kortum, F., Klünder, J., Schneider, K., 2019. Behavior-driven dynamics in agile development: The effect of fast feedback on teams. In: 2019 IEEE/ACM International Conference on Software and System Processes. ICSSP, IEEE, pp. 34–43.
Kuutila, M., Mãntylã, M.V., Claes, M., 2020. Chat activity is a better predictor than chat sentiment on software developers productivity. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. pp. 553–556.
Kuutila, M., Mäntylä, M.V., Claes, M., Elovainio, M., 2017. Reviewing literature on time pressure in software engineering and related professions: computer assisted interdisciplinary literature review. In: 2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion). IEEE, pp. 54–59.
Kuutila, M., Mäntylä, M., Claes, M., Elovainio, M., Adams, B., 2021. Individual differences limit predicting well-being and productivity using software repositories: a longitudinal industrial study. Empir. Softw. Eng. 26 (5), 1–30.
Laanti, M., 2013. Agile and wellbeing–stress, empowerment, and performance in scrum and kanban teams. In: 2013 46th Hawaii International Conference on System Sciences. IEEE, pp. 4761–4770.
Lander, A., Berg, C., Landry, H., Cagle, K., Chirravuri, L., Cooney, L., Moore, M., Robinson, M., Nair, N., Choudhary, P., Fewell, P., Mayilsamy, P., Nagappan, R., Kennady, V., Harris, V., 2016. Introduction to agile 2. https://agile2.net/.
Landowska, A., 2013. Affective computing and affective learning - methods tools and prospects. EduAkcja. Mag. Edukac. Elektron. 16–31.
Lenberg, P., Feldt, R., Wallgren, L.G., 2015. Behavioral software engineering: A definition and systematic literature review. J. Syst. Softw. 107, 15–37.
Lindsjørn, Y., Sjøberg, D.I., Dingsøyr, T., Bergersen, G.R., Dybå, T., 2016. Teamwork quality and project success in software development: A survey of agile development teams. J. Syst. Softw. 122, 274–286.
Luong, T.T., Sivarajah, U., Weerakkody, V., 2021. Do agile managed information systems projects fail due to a lack of emotional intelligence? Inf. Syst. Front. 23 (2), 415–433.
Madsen, S., Matook, S., 2010. Conceptualizing interpersonal relationships in agile IS development. In: ICIS 2010 Proceedings-Thirty First International Conference on Information Systems. Association for Information Systems.
Marshall, A., Gamble, R.F., Hale, M.L., 2016. Outcomes of emotional content from agile team forum posts. In: Proceedings of the 1st International Workshop on Emotion Awareness in Software Engineering. pp. 6–11.
Meier, A., Kropp, M., Anslow, C., Biddle, R., 2018. Stress in agile software development: practices and outcomes. In: International Conference on Agile Software Development. Springer, Cham, pp. 259–266.
Mishra, D., Mishra, A., Ostrovska, S., 2012. Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. Inf. Softw. Technol. 54 (10), 1067–1078.
Mulder, H., 2020. Emotional Intelligence and Leadership Emergence in Agile Team Meetings (Bachelor's thesis). University of Twente.
Murgia, A., Tourani, P., Adams, B., Ortu, M., 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In: Proceedings of the 11th Working Conference on Mining Software Repositories. pp. 262–271.
Nagappan, R., 2021. The missing interface between agile and leadership. https://medium.com/swlh/the-missing-interface-between-agile-and-leadership-3ac73050b783.
Noordewier, M.K., Breugelmans, S.M., 2013. On the valence of surprise. Cogn. Emot. 27 (7), 1326–1334.
Novielli, N., Calefato, F., Lanubile, F., 2018. A gold standard for emotion annotation in stack overflow. In: 2018 IEEE/ACM 15th International Conference on Mining Software Repositories. MSR, IEEE, pp. 14–17.
Novielli, N., Serebrenik, A., 2019. Sentiment and emotion in software engineering. IEEE Softw. 36 (5), 6–23.
Ostberg, J.-P., Graziotin, D., Wagner, S., Derntl, B., 2020. A methodology for psycho-biological assessment of stress in software engineering. PeerJ Comput. Sci. 6, e286.
Parrott, W.G., 2001. Emotions in Social Psychology: Essential Readings. psychology Press.
Pedrycz, W., Russo, B., Succi, G., 2011. A model of job satisfaction for collaborative development processes. J. Syst. Softw. 84 (5), 739–752.
Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12. pp. 1–10.
Petersen, K., Wohlin, C., 2009. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. J. Syst. Softw. 82 (9), 1479–1490.
Plain, C., 2007. Build an affinity for KJ method. Qual. Prog. 40 (3), 88.

Plutchik, R., 1991. The Emotions. University Press of America.
Plutchik, R., 2001. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. Am. Sci. 89 (4), 344–350.
Ptaszynski, M., Dybala, P., Shi, W., Rzepka, R., Araki, K., 2009. Towards context aware emotional intelligence in machines: computing contextual appropriateness of affective states. In: Twenty-First International Joint Conference on Artificial Intelligence.
Rodríguez, P.M., Del Pino, D., Alvaredo, R., 2011. De lo psicológico a lo fisiológico en la relación entre emociones y salud. Rev. Psicol. Cient. com 17.
Russell, J.A., 1980. A circumplex model of affect. J. Personal. Soc. Psychol. 39 (6), 1161.
Salido Ortega, M.G., Rodríguez, L.-F., Gutierrez-Garcia, J.O., 2020. Towards emotion recognition from contextual information using machine learning. J. Ambient Intell. Humaniz. Comput. 11 (8), 3187–3207.
Sánchez-Gordón, M., Colomo-Palacios, R., 2019. Taking the emotional pulse of software engineering—A systematic literature review of empirical studies. Inf. Softw. Technol. 115, 23–43.
Schmidt, A., Beigl, M., Gellersen, H.-W., 1999. There is more to context than location. Comput. Graph. 23 (6), 893–901.
Schneider, K., Klünder, J., Kortum, F., Handke, L., Straube, J., Kauffeld, S., 2018. Positive affect through interactions in meetings: The role of proactive and supportive statements. J. Syst. Softw. 143, 59–70.
Scott-Hill, B., Anslow, C., Ferreira, J., Kropp, M., Mateescu, M., Meier, A., 2020. Visualizing progress tracking for software teams on large collaborative touch displays. In: 2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). IEEE, pp. 1–5.
Serebrenik, A., 2017. Emotional labor of software engineers. In: BElgian-NEtherlands Software EVOLution Symposium. BENEVOL, pp. 1–6.
Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H., 2009. Models of motivation in software engineering. Inf. Softw. Technol. 51 (1), 219–233.
Sithambaram, J., Nasir, M.H.N.B.M., Ahmad, R., 2021. Issues and challenges impacting the successful management of agile-hybrid projects: A grounded theory approach. Int. J. Proj. Manage. 39 (5), 474–495.
Soomro, A.B., Salleh, N., Mendes, E., Grundy, J., Burch, G., Nordin, A., 2016. The effect of software engineers' personality traits on team climate and performance: A Systematic Literature Review. Inf. Softw. Technol. 73, 52–65.
Stamelos, I.G., Sfetsos, P., 2007. Agile Software Development Quality Assurance. Igi Global.
Sutherland, J., Harrison, N., Riddle, J., 2014. Teams that finish early accelerate faster: a pattern language for high performing scrum teams. In: 2014 47th Hawaii International Conference on System Sciences. IEEE, pp. 4722–4728.
Tessem, B., 2014. Individual empowerment of agile and non-agile software developers in small teams. Inf. Softw. Technol. 56 (8), 873–889.
Tripp, J.F., Riemenschneider, C., Thatcher, J.B., 2016. Job satisfaction in agile development teams: Agile development as work redesign. J. Assoc. Inf. Syst. 17 (4), 1.
Uribe, E.H., Ayala, L.E.V., 2007. Del manifiesto ágil sus valores y principios. Sci. Tech. 13 (34), 381–386.
Vishnubhotla, S.D., Mendes, E., Lundberg, L., 2020. Investigating the relationship between personalities and agile team climate of software professionals in a telecom company. Inf. Softw. Technol. 126, 106335.
Wood, S., Michaelides, G., Thomson, C., 2013. Successful extreme programming: Fidelity to the methodology or good teamworking? Inf. Softw. Technol. 55 (4), 660–672.
Xu, P., Wang, T., 2007. Emotion in decision making under uncertainty and cognitive control: a computational framework. In: Third International Conference on Natural Computation (ICNC 2007), Vol. 2. IEEE, pp. 149–154.
Zheng, Z., Wang, L., Cao, Y., Zhuang, Y., Tao, X., 2019. Towards non-invasive recognition of developers' flow states with computer interaction traces. In: 2019 26th Asia-Pacific Software Engineering Conference. APSEC, pp. 300–307.

Martín G. Salido O. is a Ph.D. student in Engineering Science at Instituto Tecnológico de Sonora, Mexico. His research interests include agile methodologies, affective computing, and software engineering.

Gilberto Borrego obtained his Ph.D. degree from the Autonomous University of Baja California in 2018. He worked as software engineer from 2002 to 2005 and from 2007 to 2014 in different Mexican and international companies. Since 2019, he has been a researcher-professor at the Instituto Tecnológico de Sonora, and he is member of the Networks and Collaborative Systems research group in the same institute. His research interests are agile software development and knowledge management.

Ramón R. Palacio was born in Navojoa, Mexico in 1977. He obtained his Ph.D. degree from the Autonomous University of Baja California in 2010. He practiced software engineering from 1999 to 2000 in a national company. Since 2003, he is a researcher-professor at the Instituto Tecnológico de Sonora where he is a member of the Networks and Collaborative Systems research group. He conducts studies to gain a better understanding of current work practices and, based on this understanding, designs, develops and evaluates appropriate technologies for various work environments.

Luis-Felipe Rodríguez received a Ph.D. in Computer Science from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV-IPN), campus GDL, Mexico. He is currently a full time Professor of Computer Science at Instituto Tecnológico de Sonora, México. His research interests include cognitive computing, affective computing, affective human–computer interaction, and decision support systems.