IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 22, 2021, accepted August 2, 2021, date of publication August 5, 2021, date of current version August 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102663

Understanding the Impact of Emotions on the Quality of Software Artifacts

KHALED M. KHAN, (Senior Member, IEEE), AND MOUTAZ SALEH
Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Corresponding author: Khaled M. Khan (k.khan@qu.edu.qa)

This work was supported by Qatar National Research Fund (a member of Qatar Foundation) through the National Priorities Research Program (NPRP) under Grant NPRP-09-079-1-013.

ABSTRACT This paper proposes a framework for investigating the impact of emotions on the quality of software artifacts and portrays some observations captured during the development of several software engineering projects. The paper uses eight primary emotions and their intensity levels defined in Plutchik's *Wheel of Emotions*. This study utilizes two major artifacts in the development environment; the software artifacts and the communication artifacts. The findings show that most developers experienced *optimism* in their work, felt *joy*, and found *interest* in the project. It also reveals that the most three influencing negative emotions are *distraction*, *anger* and *boredom.* Almost 20% of the developers were distracted during the project activities. Some developers also experienced anger with other team members. Nearly one-fifth of the participants were bored or lost interest in the project. These results confirm the findings of other similar research. However, we have also found that this claim does not always hold in some cases. Positive emotions do not always contribute to good quality artifacts. Similarly, negative emotions sometimes result in a positive impact on the quality of artifacts. The obtained results show how various kinds of emotions impact differently on the quality of software artifacts.

INDEX TERMS Software artifacts, wheel of emotions, communication artifacts, developers' emotions, software quality.

I. INTRODUCTION

One of the most commonly ignored factors of software development is the emotions of software engineers. Various events, stimuli, and situations trigger the emotions of software developers. Like other professions, software developers' emotional aspect considerably affects the quality and productivity of developers [1], [2]. It is argued in [1], [3], [39] that the happiness of software developers influences the quality and productivity of software products.

Emotions can be negative such as anger, disgust, remorse; or positive such as joy, trust, optimism. The emotional-display behaviour of an individual depends on the context that the individual is in [4]. Software developers do not always express their emotions in the communication channels; some suppress their emotions in the workplace. Regardless of negative, positive, suppressed or expressed emotions, these may have an impact on the quality of software artifacts that developers produce. Emotions felt by software developers contribute to the quality of software artifacts and their overall productivity [2], [3], [5].

To improve the software development process in terms of the quality of software products, practitioners need to understand the landscape of emotions in this profession. Early identification of negative emotional symptoms of software developers could prevent a negative impact on the quality of their software products. This paper attempts to aid in understanding impact of emotions on the quality of software artifacts. Our aim is to investigate: (i) if the emotion of developers and various intensity levels of their emotion play any role in the quality of software artifacts; (ii) if the correlation between emotions and quality is always symmetric. That is, if negative emotion always contributes to bad quality artifacts, and if positive emotion has always an impact on good quality artifacts; and (iii) if causes or sources of various emotions affecting the quality of artifacts are mostly external to the development environments or personal. To understand these three, we first capture emotions from various communication

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

documents, identify quality variations of software artifacts, and correlate the captured emotions with the perceived quality of software products.

Emotion detection can be developed from coarse-grained to more fine-grained levels [6]. In general, coarse-grained level detection categorizes emotions into positive and negative polarities. Whereas fine-grained level detection refers to classifying emotions into many cognitive states or primary emotions such as happiness, sadness, joy, anger, etc. In this paper, we plan to go one step further of the fine-grained level. That is, we attempt to capture the intensity level of each of these primary emotional states.

The topic of emotion is vast in psychology. In sentiment analysis [7], emotion detection attempts to identify the attitudes as well as the mental state of team members toward their specific development tasks. In this study, we adapt Plutchik's *Wheel of Emotions* [8] containing eight primary emotions polarized into positive and negative emotions, each with varied intensity levels, and combination. To measure and evaluate the emotions of software developers, we use the Positive and Negative Affect Schedule (PANAS) [9] assessment instrument. The paper is expected to portray some observations captured during the development of several development projects.

The remainder of the paper is organized as follows: Section 2 presents the related work. Section 3 describes our methodology, apparatus used, the control mechanisms imposed to avoid biases, validation of data and the contributions of this paper. The proposed framework is explained in Section 4. Section 5 discusses the interviews of developers. Our statistical analysis with validity is presented in Section 6. We provide a discussion of our findings in Section 7. Section 8 concludes the paper.

II. RELATED WORK

A thorough browse of the related research activities suggests that the last two decades have witnessed an increased research output in the area of emotion within the field of affective computing. Researchers use various tools and techniques to capture and analyze emotions. Different approaches are being applied to capture the emotion of individual software development team members and their impact on the quality and productivity of software artifacts. These can be classified into several categories: (1) keyword based or lexicon analysis; (2) machine learning (ML); (3) sensor-based physical and physiological signals; (4) walk-through artifacts; and (5) combination of approaches.

A. KEYWORD-BASED OR LEXICON ANALYSIS

Traditional approaches such as reported in [10]–[16], [43] are used to identify emotional keywords within a text then extract embedded emotion using rule-based dictionaries or other techniques. Sentiment analysis uses lexicons of words in the texts to extract emotions. Tools for sentiment analysis such as Senti4SD [44], SentiStrength-SE [17] based on heuristics are reported to have provided better results in analyzing emotions in software artifacts. However, it is also argued in [18] that polarity detection-centric sentiment analysis without manual support is not always sufficient in detecting reliable emotions of developers from software artifacts.

B. MACHINE LEARNING

In the machine learning approach, a model is built and trained using a large amount of data before the emotions of test data are classified. Various ML algorithms [19]–[24] such as support vector machine (SVM), Naïve Bayes, decision trees, convolutional neural network (CNN), long short-term memory (LSTM) are used. For instance, the use of machine learning, in particular SVM, was reported in [25]. The methods based on supervised learning mostly use sentiment embedding-based model using accelerated algorithm and support vector regression [26], deep learning [21]–[24], [27] maximum entropy [28], hybrid neural network [19], pattern recognition [29] to extract emotions from texts. These techniques and their combinations have emerged as a powerful tool to recognize the cognitive state of development team members. Martens and Maalej [30] utilize sentiment-analysis tools to identify users' emotions usually conveyed with different recurring patterns. In the proposed approach in [31], the authors apply machine-learning and linguistic features to classify sentiment. They report to have successfully detected both positive and negative emotions on publicly available dataset as benchmarks.

The work in [6] builds an emotion-embedding model using Tweet data and then uses a sentiment analyser to extract emotional words. The model is then trained with the extracted Tweet emotion data. One of the main drawbacks of this research is the ignoring of contextual information in the Tweet data. Similar to our research paper, this work uses Plutchik's wheel of emotions but is limited to eight basic emotion types and did not take the intensity levels of emotions into account.

C. SENSOR-BASED

Various sensor-based physical and physiological measures [32]–[34] have been used to extract emotions. These include electrical activity of the brain (EEG), electrodermal activity of the skin (EDA), electrical activity of the contracting muscles (EMG), blood volume pulse (BVP), brain signals, heart rate and heart rate variability [36]. These methods entirely depend on physiological signals of persons to capture expressed as well as suppressed emotions. The effectiveness and appropriateness of such approaches in the software development process are not very conclusive.

D. WALK-THROUGH ARTIFACTS

This approach is considered non-automatic but most effective if conducted properly and systematically. Indeed, it is argued that if humans cannot determine any emotion from software development artifacts, no tool can [37]. In [38], the authors argue that emotional clarity can identify the source of developers' emotions. Their work advocates for

TABLE 1. Summary of the existing approaches.

Approaches	Objectives	Techniques used	References
• **Keyword-based or lexicon analysis**	• Detect and identify emotional key words within texts • Extract and analyze emotions • Sentiment analysis.	• Rule-based dictionaries • Lexicons • Heuristics • Polarity detection-centric analysis	10, 11, 12, 13, 14, 15, 16, 17, 18,.
• **Machine learning**	• Classify emotions and sentiment • Extract emotions from texts • Recognize cognitive state of software developers • Identify users' emotions	• Support vector machine, Naïve Bayes, decision trees, convolutional neural network (CNN), long short-term memory (LSTM) • Accelerated algorithm, support vector regression, maximum entropy, pattern recognition	6, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31.
• **Sensor-based**	• Detect various physiological signals to capture expressed and suppressed emotions.	• Electrical activity of the brain (EEG) • Electrical activity of skin (EDA) • Electrical activity of contracting muscles (EMG), Blood volume pulse (BVP)	32, 33, 34, 36.
• **Walk-through artifacts**	• Detect emotions • Capture emotional clarity to identify the source of emotions	• Interviews • Walk-throughs of artifacts	1, 3, 38, 39.
• **Combinati on of approaches**	• Detect emotions • Identify emotional labor.	• Survey, Questionnaire. • Valence, arousal and dominance (VAD) lexicon and Senti4SD	40, 41, 42, 43, 44. 56, 57, 58.

more about the development of an emotion-aware environment that is expected to improve the developers' emotional responses in software development. In another research, Graziotin *et al.* [3] studied 42 developers with interviews and walk-throughs of artifacts to conclude that positive emotions and moods positively affect their performance and ultimately contribute to the quality and productivity of software products. The happiness of software developers can influence their ultimate output [39].

E. COMBINATION OF APPROACHES

Besides developing tools and walk-throughs to detect emotions, various approaches coupled with surveys are also considered effective research methods. Serebrenik [40] convincingly argues that software developers are subject to emotional labour, which can be identified using emotion detection techniques. The paper further stresses that necessary interventions can also be designed to tackle the challenges of emotional labour. It further argues that different types of biometrics such as textual sources, formal and informal communication, conversations, and artifacts could reflect developers' emotions. Emotional dissonance forces software developers to display emotions that might be different from what they felt. This work points out that they may voluntarily entangle their genuine emotions by suppressing their negative emotions regardless of what they feel.

Similar research was conducted by Gunsel [41] to examine the relationships among emotional labour and software quality, such as effectiveness, flexibility and responsiveness, using survey data from 62 software development projects. The researchers in [42] attempt to use a general purpose Valence, Arousal and Dominance (VAD) lexicon, and apply this to a large database of software development process. They confirm that increased emotions in terms of VAD do correlate with increased productivity in software development. Sentiment analysis is one of the techniques that researchers have been using widely to identify and recognise emotions from various sources such as texts and conversations [43]. The research reported in [44] studied mainly the polarity of emotions extracted from the texts. It argues that users usually leave traces of their emotions embedded in various texts. Another research work in [20] tries to correlate software team members' sentiment and software defects. It attempts to detect Fix-Inducing-Changes (FIC) changes that result in bugs in the software. The approach uses Senti4SD [44] to detect sentiments of these changes.

Table 1 summarises the existing research approaches used in the field of emotions of software developers. Most of the existing research focuses on primary or basic emotion types. They do not identify various intensity levels of primary emotions and their impact. On the contrary, our work uses other non-primary emotions that are simply a combination of the eight primary emotions or are derived from one or more of these. Our research attempts to capture the intensity level of the primary emotions as high intensity and low intensity to accurately study the impact of those emotions on the quality of software artifacts.

III. OVERVIEW OF THE STUDY

Twenty senior students in computer science (CS) major at Qatar University are involved in five projects to develop software systems. These projects include all phases of the software development life cycle (SDLC) such as analysis, design, implementation and testing. The duration of each project is four months. We supervised five software engineering teams, and each was having four female developers. The team members were mixed of good, average, and below-average based on their prior academic performance in five courses, namely object-oriented modeling, programming concepts, object-oriented programming, data structures, and algorithms design.

Our research methodology utilized two project artifacts: *software artifacts* and *communication artifacts*, produced by the team members to capture emotion and correlation between artifacts and captured emotion of the developers. We later conducted interviews to find out if emotions expressed in the *communication artifacts* (e.g., email, meeting minutes, project logs, reflection) impact on the quality of the delivered *software artifacts*. We maintained a Correlation Table (explained in section 4.3) of each team member to record the correlation between the emotion expressed in the communication artifacts and their software artifacts. The interviews were conducted to confirm captured emotions and their relationship with the quality of software artifacts. These are also needed to get additional information about the causes of their emotions.

The following subsections explain, (i) the objectives of our research; (ii) the types of emotion we have adapted; (iii) the apparatuses used; (iv) the control mechanisms applied to avoid biases; (v) the conducted interviews of the participants; and (vi) the outlines of the specific contributions of this paper.

A. OBJECTIVES

First, this study attempts to find out if the emotions (positive or negative) of participants impact the quality of software artifacts that they develop. We predict that the software development outputs produced by the participants are sometimes influenced by both positive and negative emotions, and their intensity levels. The main objectives of our study are:

1. Investigate if the emotions of participants have any impact on the quality of software artifacts
2. Explore if positive emotion is always responsible for good quality software artifacts; and if negative emotion always contributes to bad quality artifacts.
3. Examine if the causes or sources of various emotions affecting the quality of artifacts are primarily external to the development environments or personal.

B. PLUTCHIK's WHEEL OF EMOTIONS

Emotions are about the feelings of individuals [45], usually expressed differently. Some emotions are expressed through facial appearances, gestures, verbal or in writing. Some emotions that we experience have little change in our appearances or gestures. This study used eight discrete primary emotions, their intensity, and combination based on the Plutchik's wheel of emotions [8]. The primary emotions are *joy, trust, fear, surprise, sadness, anticipation, anger*, and *disgust.* These relate to one another in various ways. Each of these eight emotions has a polar opposite, meaning we have four pairs of emotions. In each pair, one emotion is opposite to the other. These four pairs based on eight primary emotions are *sadness* vs. *joy; anger* vs. *fear; surprise* vs. *anticipation;* and trust vs. disgust. Table 2 shows the list of the pairs of primary emotions. Plutchik did not explicitly mention *negative* or *positive* emotions. Instead, he refers to an emotion with its opposite emotion. We use these in this paper as *positive* and *negative* emotions.

Table 3 depicts two intensity levels of each primary emotion as *High intensity* and *Low intensity*; and two different combinations of two primary emotions shown as *Combination 1* and *Combination 2*. For example, in Combination 1 column, *remorse* is an emotion as a result of the combination of the two primary emotions *sadness* and *disgust* (See the first row). Similarly, *disapproval* is an emotion based on the combination of *surprise* and *sadness* primary emotions (See the second row). Each primary emotion has two possible derived emotions shown in Combination 1 and Combination 2 columns respectively in Table 3. Emotions in italics within parentheses in Combination 1 and Combination 2 columns denote primary emotions. Other non-primary emotions are simply a combination of these eight primary emotions, or are derived from one or more of these. For instance, *optimism* can be derived from *Joy* and *anticipation*; *submission* can be derived from *trust* and *fear*; *contempt* is derived from *disgust* and *anger*, and so on. Emotion can have a varying degree of intensity, such as *annoyance*, the low intensity of the primary emotion *anger*; whereas, a *rage* is the highest-level intensity of *anger*. *Grief* and *pensiveness* are the high and low intensity of the primary emotion, *sadness* respectively.

TABLE 2. Plutchik's primary emotions with polar opposite.

Positive	Negative
Joy	*Sadness*
Fear	*Anger*
Anticipation	*Surprise*
Trust	*Disgust*

C. APPARATUS

We use two major apparatuses in our study, namely *Software artifacts* and *Communication artifacts*:

Apparatus 1: Software artifacts: This includes three *project deliverables* in five weeks interval:

- The first deliverable includes *requirements analysis* such as use cases, use case specifications, activity diagrams, domain models, and data flow diagrams.
- The second deliverable involves *design artifacts* of the system, e.g. class diagrams, sequence diagrams, interface layouts, component and package diagrams.
- The third and final deliverable consists of the implementation artifacts such as programs, test results, and deployment diagrams.

Apparatus 2: Communication artifacts: The teams are required to keep all *communication artifacts*.

- The communication artifacts include (i) *email exchanges*, (ii) *discussion board*, (iii) *team-meeting minutes*, (iv) *project logs*, and (v) *the reflection of their experience with the project.*

D. CONTROL MECHANISM

The participants of this study were senior students in the computer science major. We did not disclose the purpose of this study to the participants due to three reasons:

- The study had no positive impact on them if they knew it;

TABLE 3. Plutchik's wheel of emotions with combination of primary emotions and levels of intensity.

Primary Emotion	High intensity	Low intensity	Combination 1	Combination 2
Sadness	Grief	Pensiveness	Remorse (*Disgust*)	Disapproval (*Surprise*)
Surprise	Amazement	Distraction	Disapproval (*Sadness*)	Awe (*Fear*)
Fear	Terror	Apprehension	Awe (*Surprise*)	Submission (*Trust*)
Trust	Admiration	Acceptance	Submission (*Fear*)	Love (*Joy*)
Joy	Ecstasy	Serenity	Love (*Trust*)	Optimism (*Anticipation*)
Anticipation	Vigilance	Interest	Optimism (*Joy*)	Aggressiveness (*Anger*)
Anger	Rage	Annoyance	Aggressiveness (*Anticipation*)	Contempt (*Disgust*)
Disgust	Loathing	Boredom	Contempt (*Anger*)	Remorse (*Sadness*)

- They did not have the opportunity to control or camouflage their emotions artificially; and
- We wanted to capture their natural emotions.

We avoided the participants' awareness biases by not telling our research objectives. We ensured that the participants would complete the project tasks as natural as possible without being preoccupied with our research goals. Making the participants unaware of our research goals prevented them from manipulating their emotional behavior reflecting on their project tasks. We did not ask a priori for any self-reports of their emotional experience during the project because these could be artificially refined or biased. They might tend to respond in such a manner that could reflect their artificially created positive emotions.

E. CONDUCTED INTERVIEWS OF PARTICIPANTS

At the end of the final project delivery, we conducted interviews with questionnaires representing a psychological construct. The main objectives of the interviews were to investigate the views of the participants regarding the state of their various emotions and their impact on the software products. Another aim of the interviews was to confirm the correlation between two artifacts: software artifacts and communication artifacts. We use Positive and Negative Affect Schedule called PANAS measurement instrument to record and validate emotion data from the interview. The instructor populated the questionnaires with data during the interview session with the individual participant. We acknowledge the fact that it is complex and challenging to observe the psychological construct of the participants.

F. CONTRIBUTIONS

This study results in the following specific contributions.

1) The findings confirm the earlier studies, such as in [1], [3], [39], [56]–[58] that in most cases, positive emotions of software developers have positive impact on the quality of software artifacts they deliver and their productivity.
2) This research identifies that positive emotions do not always contribute to good quality software artifacts, reported by other researchers. Similarly, negative emotions do not always result in bad quality software artifacts [39], [56], [57]. Sometimes there is no symmetric correlation between emotions and their impact on the quality of software products.
3) This study ensures that the proposed layered framework with four distinct tasks to capture the impact of emotions on the quality of software artifacts is a viable Walk-Through Artifact process.
4) The data show that software developers' most dominating positive emotions are *joy*, *interest* in the project, and *optimism*; and the most negative emotions are *distraction*, *anger*, and *boredom*. Optimism is a combination of two primary emotions, whereas interest, distraction and boredom are the low intensity of primary emotions.
5) The results confirm that most software developers do not hide their positive emotions, although they try to suppress negative emotions.
6) The findings identify some causes and sources of expressed emotions of software developers.
7) This research proposes an Emotion-Landscape to graphically capture the relationship of expressed and suppressed emotions with the quality of artifacts with various color combinations.
8) The study demonstrates that communication artifacts used by software developers are essential sources for finding emotions, and that can be used along with the software quality artifacts to detect the correlations between quality products and emotions.

IV. PROPOSED FRAMEWORK

We propose a framework to study the impact of emotions on five software development projects. The framework is based on a layered structure and consists of four tasks as depicted in Figure 1: (i) Analyzing software artifacts (blue circle), (ii) Mining emotional features (yellow circle), (iii) Developing correlation (green circle), and (iv) Reasoning sources (red circle). The blue-colored texts in these four-layered in the framework denote possible outcomes of the corresponding task; however, these are not exhaustive. For example, developing correlation can result in positive, negative, unknown, or no impact on the quality of software artifacts.

A. ANALYZING SOFTWARE ARTIFACTS

The first task analyzes software artifacts delivered by each developer to identify the quality of the artifacts relative to

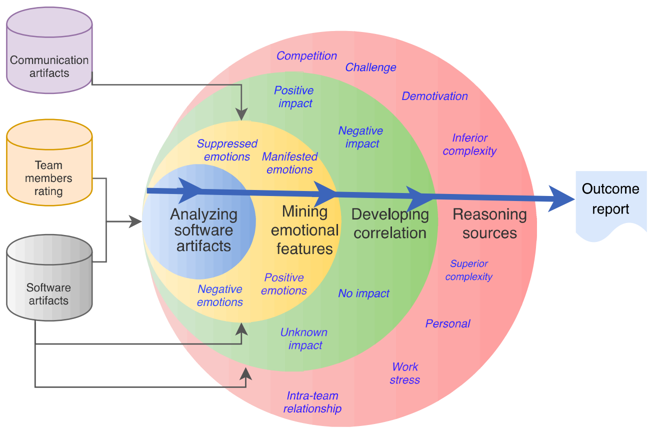


FIGURE 1. **A framework to capture the impact of emotions on the quality of software artifacts.**

the skills of each team member. We used three parameters to determine the quality of artifacts:

- *Completeness (C1)* means a delivered artifact has all the required components. For example, all use cases included in the use case diagram are defined, and each use case specification includes all actor's actions and the system's response with alternative flows if needed. These indicate that this deliverable is complete. Another example of completeness is that an activity diagram modeling a use case must have a start and an end process with all required intermediate processes and decision points.
- *Consistency (C2)* requires the artifacts logically related to other artifacts. For instance, the sequence diagrams (SD) reflect the class relationships specified in the class diagram. The messages in SD must be consistent with the methods defined in the corresponding classes, and the association relationship between the types is consistent with the messages passed between objects in the SD.
- *Correctness (C3)* denotes the accuracy of the artifacts. For example, the proposed classes in the design class diagram are adequate to capture the objects manipulated in the system. The sequence and flow of messages in the SD is correct in relation to the response of the system in the use case specification normal and alternative scenario. Another example is, the use cases in the use case diagram and their relationships with the actors and other uses cases represent the system's functionalities correctly.

Each of the above three quality attributes is assigned a score in 5-point Likert scale, where 1 represents poor quality, 3 stands for average quality, and 5 represents the highest quality. The quality of software artifacts may reflect the impact of emotions of the developers. For instance, if a design document produced by a competent team member is of poor quality, there is a possibility that her emotion may have contributed to such low quality of the design. Indeed, there is a need for confirmation if the developer was somehow emotionally influenced while designing this specific component. In that case, we explore the communication artifacts in the next task to identify the emotional vocabulary of the developers.

B. MINING EMOTIONAL FEATURES

The intra-team communications such as email exchanges, discussion boards, team-meeting proceedings, and reflection of an individual team member may have captured vital vocabularies related to the participants' various emotions. To track causes for a team member's performance, this task explores the scripts available from the communication artifacts. This task also reviews the quality of software artifacts in light of the emotional features identified from the communication artifacts using the walk-through approach.

Generally, there is a symmetric relationship between the software artifacts and the communication artifacts; meaning we can recognize the quality of software artifacts resulted in emotions expressed in the communication artifacts or vice versa. Similarly, we can also identify the impact of emotions on the software artifacts from the emotions expressed in the communication artifacts. However, this is always not the case as we see later.

Emotions are often embedded in messages that the team members use to communicate [46]. These messages are quite valuable to detect the emotions of the message writers of the team. These short messages usually span several sentences. Emotions expressed in these messages can represent the happiness, sadness, anger or frustration of the message senders. The responses to these messages by other team members can also reinforce or even challenge the underlying emotions. Messages and responses can establish an actual emotional state of the message sender in the context of software development.

Various tools and techniques such as Senti4SD [47], discrete emotions detection [48], the valence-arousal-dominance-based detection [42] are available to mine emotions using sentiment lexicon [43], keyword-based features and dialogues embedded in the artifacts. These extracted emotions, positive or negative, could have contributed to the quality of the software artifact, such as completeness, consistency and correctness. Once we identify and build a link between the quality of software artifacts and emotions, we develop correlation between them.

C. DEVELOPING CORRELATION

The inputs to this task are the software artifacts and the outcomes of the previous task. If the produced artifact is below-average quality prepared by a developer with an above-average rating, it demonstrates that she underperformed due to some reasons. Emotions could be one of such reasons. This task links the tracked emotion-oriented vocabularies with the quality of the artifacts.

To perform this task, we used a two-dimensional table called 'Correlation Table' to capture the relationship between

TABLE 4. A sample format of the correlation table.

	Developer name: GPA of 5 courses:						Project ID: Rating:					Date: Comment:				
	Analysis Artifacts															
		Use case diagram			Use case spec.			Activity diagrams			Domain model			Data flow diagram		
		C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
Email exchanges	Ref. no															
	Score															
	Emotion															
Discussion board	Ref. no															
	Score															
	Emotion															
Team-meeting proceedings	Ref. no															
	Score															
	Emotion															
Project log	Ref. no															
	Score															
	Emotion															
Reflection	Ref. no															
	Score															
	Emotion															

the communication artifacts and the software artifacts for each developer. The columns represent the components of the software artifacts, and the rows represent each specific type of communication artifact that a developer is involved in. If a correlation exists between them, we record a reference number for each artifact, the score in terms of quality of three attributes (C1, C2, C3), and the name of the emotion (i.e. anger, fear, sadness, etc.) in the intersection of the corresponding column and the row. A sample template of the 'Correlation Table' is provided in Table 4. We only show the systems analysis artifacts (use cases, activity diagram, domain model, data flow diagram) in the table for simplicity. C1, C2, and C3 represent Completeness, Consistency, and Correctness, respectively. Once the Correlation Table is complete, we describe details of the detected emotion with the reference number in an explanation document. Each developer's Correlation Table is accompanied by an explanation document to capture additional information with a narrative.

D. REASONING SOURCES

Based on the 'Correlation Table' (Table 4), this post-development activity attempts to find out the causes of the identified emotions. This task uses *informal conversations* (interviews) of individual developer with a questionnaire. The source of the emotion can be competition, challenges, demotivation, inferior complexity, superior complexity, personal, work stress, team relationships, etc. After the delivery of all milestones, we designed a questionnaire for them based on PANAS measurement instrument. The main objective of the interview was to find out if: (i) a participant had undergone any emotional episode during the project tasks; (ii) the interviewee noticed any emotion of other team members; (iii) the correlation is correct; (iv) the participant has any additional information about their expressed emotion; and (v) they suppressed any emotion.

V. INTERVIEWS OF PARTICIPANTS

In the project-exit interviews, the twenty developers were asked individually to outline their emotional conditions and explain in detail what they believed could have caused them to experience the emotions and whether and how their emotion influenced their software development activities. The interviews are also intended to capture the mapping of the emotion of individual developers to the specific quality of the software artifacts. However, the exact mapping between the individual student and the particular quality of an artifact appeared quite challenging. Therefore, our main objective is to investigate if the participant's emotion during the development time has any impact on the quality of the software artifacts. The interviews were based on PANAS with ten positive emotions and ten negative emotions, in shown in Tables 5 and 6, respectively.

TABLE 5. Ten positive emotions used in the interviews.

	1 Not at all	2 A Little	3 Moderately	4 Quite a Bit	5 Extremely
Joy	2	4	3	4	7
Anticipation	12	2	5	0	1
Trust	9	5	2	2	2
Interest	0	1	3	5	11
Serenity	16	2	2	0	0
Acceptance	12	5	2	1	0
Optimism	2	1	5	4	8
Love	11	4	3	1	1
Submission	17	2	1	0	0
Admiration	15	3	2	0	0

TABLE 6. Ten negative emotions used in the interviews.

	1 Not at all	2 A Little	3 Moderately	4 Quite a Bit	5 Extremely
Anger	2	2	4	8	4
Disgust	15	3	1	0	1
Sadness	14	5	1	0	0
Distraction	2	0	3	6	9
Pensiveness	9	6	2	2	1
Boredom	3	4	3	5	5
Annoyance	11	0	2	1	6
Aggressiveness	11	4	5	0	0
Contempt	8	4	5	1	2
Disapproval	4	5	6	4	1

Table 5 is comprised of 10 positive emotional items, with three items measuring Plutchik's primary emotions (*Joy, Anticipation,* and *Trust*); one item for High intensity (*Admiration*); three items for Low intensity (*Interest, Serenity*, and *Acceptance*); and remaining items for Combinations 1 & 2 emotions as depicted earlier in Table 3. Table 6 is comprised of 10 negative emotions, with three items measuring Plutchik's primary emotions (*Anger, Disgust*, and *Sadness*); four items for Low Intensity emotions; and remaining three are for Combinations 1 & 2. Each item is scored on a five-point Likert Scale, ranging from 1 = *Very Slightly or Not at all*, to 5 = *Extremely*, to measure the extent to which the affect has been experienced during the project work.

VI. STATISTICAL ANALYSIS VALIDITY

Before starting our data analysis, we qualitatively cleaned the collected data set by removing any meaningless participants' responses, i.e. irrelevant responses to the research questions. We applied the Partial Least Squares (PLS) analysis using SmartPLS 3.0 software [49], and we tested it with a two-tier procedure, namely the measurement model, and the structural model [50]. Figure 2 shows the proposed experimental model's significant paths after running the bootstrap 5,000 times.

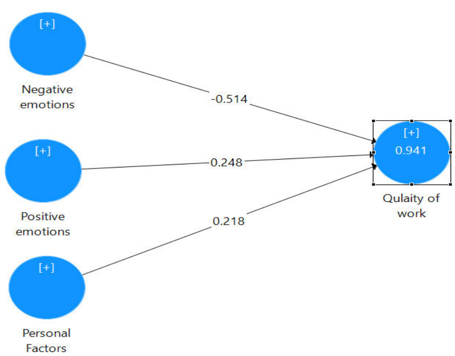


FIGURE 2. Proposed experimental model.

A. MEASUREMENT MODEL RESULTS

We tested the reliability and validity of the measurement model based on the assessment in [51]. As shown in Table 7, four parameters were used for each construct: indicator loading, composite reliability (CR), Cronbach (α), and the average variance extracted (AVE). The obtained test results clearly show that all factor loading values and CR index exceeded the 0.7 cut-off point suggested in [52]. These outcomes indicate that the model passed the construct reliability test. To verify the required model's convergent validity in [53], our test shows that all AVE values are above the cut-off point of 0.5. For the discriminant validity [50], Table 8 exhibits that each latent variable has more variance with its indicator on the diagonal line than with any other latent variable.

TABLE 7. Psychometric properties of the measurement model.

Measure	Indicator loading	Composite Reliability (CR)	Cronbach α	AVE
Negative emotions		0.98	0.975	0.908
	0.960			
	0.955			
	0.961			
Personal Factors		0.981	0.971	0.945
	0.969			
	0.976			
	0.971			
Positive emotions		0.972	0.957	0.921
	0.958			
	0.969			
	0.952			
Quality of work		0.968	0.951	0.91
	0.948			
	0.959			
	0.955			

TABLE 8. Latent variable correlations.

	Negative emotions	Personal Factors	Positive emotions	Quality of work
Negative emotions	0.953			
Personal Factors	-0.962	0.972		
Positive emotions	-0.966	0.963	0.965	
Quality of work	-0.964	0.951	0.952	0.954

B. STRUCTURAL MODEL RESULTS

To study the psychometric properties of the structural model, Table 9 represents the coefficient of determination (R2),

TABLE 9. Psychometric properties of the structural model.

	R^2	Q^2	Communalities
Negative emotions	0.981	0.789	0.908
Positive emotion	0.964	0.751	0.945
Personal Factors	0.923	0.717	0.921
Quality of work	0.941	0.701	0.910

relevance or predictive validity (Q^2) and communality for all constructs. The R^2 for our dependent variable showed 0.941 which is considered high according to [50]. The Q^2 scores are above zero indicating that they have great predictive relevance. The communality values are also above the cut-off point of 0.5 [49]. This confirms how much variance each construct shares with all other variables included in the model.

To assess how important each construct is for the model, Table 10 reports the effect size (f^2). As shown, f^2 values of the model are above the critical point of zero [50] and range from small to medium.

TABLE 10. Effect size (f^2).

	f^2	Effect size
Negative emotions	0.234	Medium
Positive emotion	0.047	Small
Personal Factors	0.054	Small

The proposed model's significant path coefficients and moderates R^2, Q^2 and f^2 indicate the strong explanatory power, high predictive relevance and applicability of the chosen variables; therefore, it is appropriate now to provide our results and discussion of the model's findings.

VII. RESULT AND DISCUSSIONS

In this section, we summarize the results of our investigation. Our analysis mainly evolved around three perceptions:

a) Impact of positive and negative emotions on software quality
b) Consequences of suppressing and expressing emotions on software artifacts, and
c) Common causes of identified emotions during software development.

A. RESULTS ON POSITIVE EMOTIONS

Figure 3 depicts that the developers did experience various positive emotions during the development time. The results show that most developers experienced *optimism* in their work, felt *joy*, and found *interest* in the project. The study also revealed that interest of the developers in the project is the most prominent compared to other emotions. Figure 4 shows

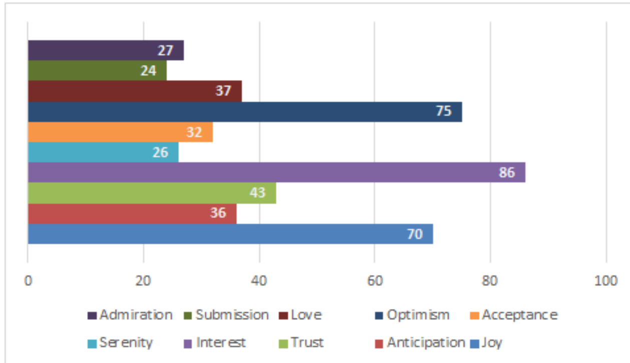


FIGURE 3. Percentages of positive emotions experienced.

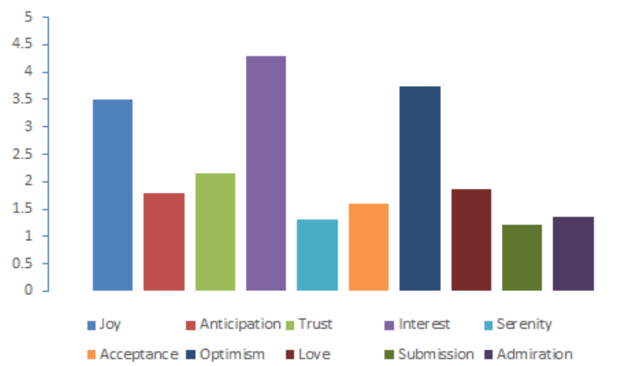


FIGURE 4. Mean values for positive emotions experienced.

the specific mean values for positive emotions experienced by the developers. However, the study did not investigate if the same developers experienced all these three emotions.

Dominating Positive Emotions:

- **Interest**: The developer likes the project so much that it sparks curiosity to achieve goals –an urge to explore more. This is the low intensity of primary positive emotion *anticipation*.
- **Joy:** This primary positive emotion displays a developer's feeling of pleasure due to an achievement or success.
- **Optimism:** This emotion is a combination of two primary positive emotions *joy* and *anticipation*. It prompts the developer to believe that things would get better.

B. RESULTS ON NEGATIVE EMOTIONS

Figure 5 captures the negative emotions that the developers experienced. The most three influencing negative emotions are *distraction*, *anger* and *boredom*. The data in Figure 6 confirm the consistency with the information in the correlation table. Almost 20% of developers were distracted during the project activities. Some developers also experienced anger with other team members. Nearly one-fifth of participants were bored or lost interest in the project. However, we are not sure if the developers who experienced distraction are the same who felt anger and boredom.

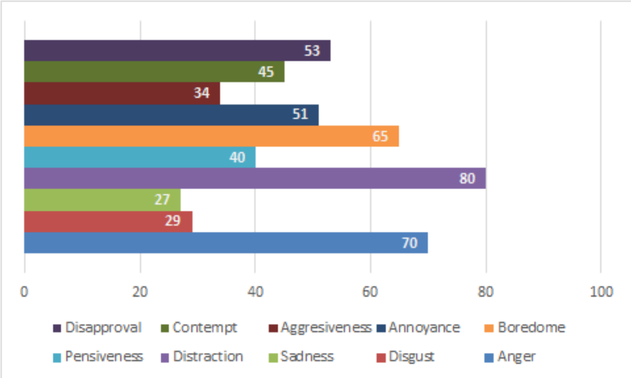


FIGURE 5. Percentages of negative emotions experienced.

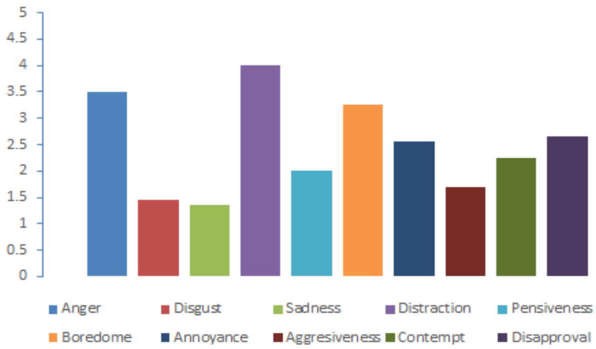


FIGURE 6. Mean values for negative emotions experienced.

Dominating Negative Emotions:

- **Distraction:** This is low intensity of the primary negative emotion *surprise*. It is an internal tension of the developers for possible failure outcomes of the project. They may experience an urge even to escape the project when they have this emotion.
- **Anger:** This is a primary negative emotion that a developer can express due to various reasons. It can be a conscious or unconscious attempt to force things to go her way or make other developers know about her unhappiness.
- **Boredom:** It is a feeling of helplessness and can happen due to lack of skill, progress, or unhappiness. One does not get what he/she wants; other developers in the team do not do what she wants, etc. This is low intensity of the primary negative emotion *disgust*.

C. DATA ANALYSIS

We analyzed data captured from three sources: *Correlation Table, Interviews with PANAS*, and *Informal Conversations*. The analysis in Figure 7 shows that some developers either expressed or suppressed their emotions. Expressed emotions are normally associated with subjective experience and context that can result in facial expression, cognitive process, behavioral changes, and other physiological reactions [4]. Expressed emotions can be positive or negative. The broaden-and-build theory developed in [54] describes how positive emotions lead to novel and positive behavior of individuals.

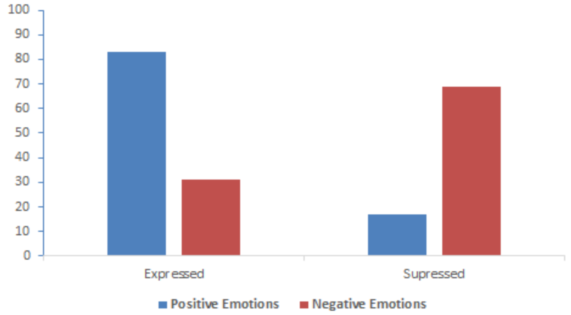


FIGURE 7. Average distribution of both expressed and suppressed positive vs. negative emotions.

Kim *et al.* [4] explored how the emotions of individuals are displayed based on context.

Sometimes emotion could be expressed (displayed) or suppressed (managed). It is interesting to note that expressed emotions such as joy and anger of some developers did not impact on the quality of their software artifacts. Suppressed emotions are difficult to be identified and traced. Suppressed emotions are not usually felt by other developers and are not visible in the communication artifacts [38]. Therefore, it is initially difficult to conclude whether the quality of a software artifact has any impact due to suppressed emotions or not. The post-development interviews and informal conversations with each developer revealed the following interesting insights.

- A few of them had experienced momentary *distraction*, *anger* and *boredom* during the project but *did not* express them. A post-development analysis of their communication artifacts did not detect any such emotions of these particular developers.
- Every developer later confirmed that they did not suppress positive emotions. However, some deliberately suppressed momentary negative emotion, in particular, their anger from other team members. Suppressed emotions are sometimes the result of emotional dissonance, meaning a gap between the emotion genuinely felt and emotions expressed [40]. Emotional dissonance has a negative impact on the quality of software artifacts as reported in [41].
- Almost half of the twenty developers suppressed their anger and boredom using surface acting to manage these negative emotions while expressing positive emotions towards others. They later confirmed that their suppressed anger somehow contributed to the quality of their products negatively.
- It was also found that expressed emotions are more prominent when some developers were more reluctant or willing to accept criticisms of other colleagues, or they showed or lacked empathy towards their fellow team members.
- A few developers disclosed that their emotions were sometimes expressed without their control. This phenomenon confirms the findings in [38].
- Another interesting observation is that a couple of developers expressed their anger consciously in a controlled

manner. However, most developers suppressed negative emotions like anger and boredom while amplified joy and interest towards others. This observation confirms the results reported in 55].

Based on these findings, we propose a graphical representation, called *Emotion-Landscape*, capturing expressed and suppressed emotions of both polarities (negative and positive) and their impact. Figure 8 depicts the proposed Emotions Landscape. The left rectangle (blue) [A, B, E, F] and the right rectangle (pink) [B, C, D, E] represent positive and negative emotions, respectively. The division of these two with the diagonal red dashed line produces two triangles: Expressed emotions [A, C, F] and suppressed emotions [C, D, F]. The expressed emotion triangle covers approximately two-thirds of the positive and one-third of the negative emotions. It signifies that software developers expressed their positive emotions more easily compared to negative emotions. Whereas the negative emotions were often suppressed.

Figure 8 also depicts that the expressed positive emotion (blue part of the [A, C, F] triangle) has a more positive impact (overlapping with the large green circle) on the software artifacts than negative impacts (blue rectangle overlapping less with the orange circle). Whereas the suppressed negative emotion (pink part of the [C, D, F] triangle) has a more negative impact (overlapping with the large orange circle) on software artifacts. Suppressed positive emotions (relatively lesser part of the green area in the triangle [C, D, F]) have a low positive impact compared to expressed positive emotions. This figure does not represent any precise quantitative value; it simply summarizes the observations symbolically.

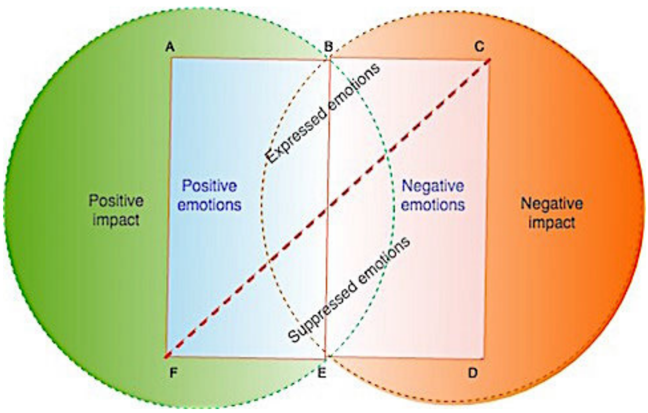


FIGURE 8. An emotion-landscape of a software project.

D. DISCUSSIONS

The findings of this research confirm the results reported in several similar but not identical studies using a walk-through approach in combination with survey [1], [3], [39], [56]–[58]. Our study shows the following dominating positive emotions expressed by the developers:

- *Joy*, a primary positive emotion
- *Optimism*, a combination of two primary positive emotions, *joy* and *anticipation*, and
- *Interest*, a low intensity of primary positive emotion *anticipation*.

These emotions of software developers always contribute to good quality software artifacts. Similarly, the following expressed negative emotions are the most prominent ones captured in our study.

- *Anger*, a primary negative emotion
- *Boredom*, a low intensity of primary negative emotion, *disgust,* and
- Distraction, a low intensity of primary negative emotion, *surprise*.

These have negatively affected the software products. A survey of 49 participants shows that positive emotion of software developers results in high productivity, and negative emotion like frustration contributes to low productivity [1]. Graziotin and his teams conducted several studies [3], [39], [56], [57] to confirm the claims of large software development companies that software developers' positive emotion directly correlates with software productivity and quality.

A study reported in [58] conducted much earlier also backs the general idea that positive emotion of software developers is responsible for better performance, such as debugging of software. In a research project with 42 unhappy and 32 happier software developers, the results support the notion that the negative emotion of unhappy developers contributes to low productivity and low performance [39]. A similar study reported in [56] also confirmed that unhappy developers with negative emotion results in low cognitive performance and low productivity in software development. In [57], 43 computer science students participated in two studies, one for creative performance and another for the analytical performance of software development tasks. Both studies support the claim that happy developers are indeed better programming problem solvers.

However, none of the studies reported that negative emotion also sometimes contribute to positive outcomes in software development. Our study has found that the claim of negative emotion contributing to low-productivity and low-quality software does not always hold in some cases.

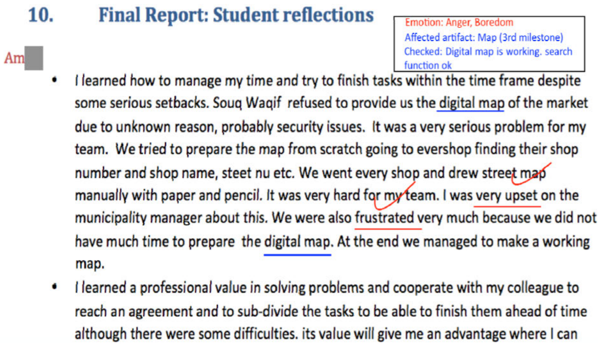
Expressed positive emotions do not always contribute to the positive quality; these can also have negative, unknown, or no impact at all on the quality of the artifacts.

For instance, a developer was feeling happy (*joy*) due to a recent change in her situation. Interestingly, this positive emotion contributed negatively to the quality of her artifacts because she was so preoccupied with her happiness (joy) that it led her to produce inconsistent and incorrect software artifacts. She did not devote the required skills to her assigned tasks due to her over-engagement in happiness. Conversely, the joy and anger of some developers expressed in the communication artifacts contributed no impact on the quality of their software artifacts.

Another observation is that negative emotions like anger and boredom may not always contribute to negative impact; rather, these can instigate a challenge in the mind of the software developer to do better. In one case, the anger of a

TABLE 11. Summary of the comparison with other similar research results.

	Paper reference [1]	Paper reference [3]	Paper reference [39]	Paper reference [56]	Paper reference [57]	Paper reference [58]	Our research
Positive emotion contributes to positive outcomes	✔	✔	✔	✔	✔	✔	✔
Negative emotion contributes to negative outcomes	✔	✔	✔	✔	✔	✔	✔
Captured primary emotions	✔		✔	✔	✔		✔
Captured intensity level of emotions							✔
Negative emotion can also contribute to positive outcomes							✔
Positive emotion can also contribute to negative outcomes							✔
Identification of causes of emotions of developers							✔

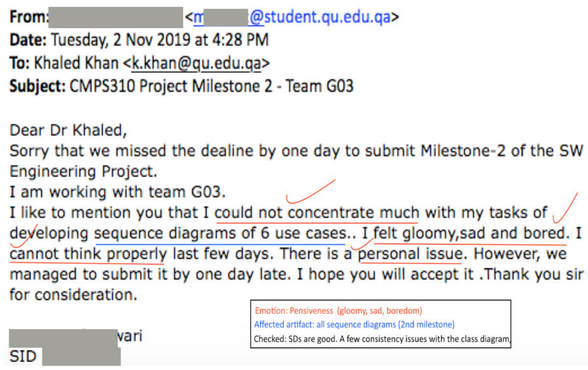

10. Final Report: Student reflections

Emotion: Anger, Boredom
Affected artifact: Map (3rd milestone)
Checked: Digital map is working. search function ok

Am[redacted]

- I learned how to manage my time and try to finish tasks within the time frame despite some serious setbacks. Souq Waqif refused to provide us the digital map of the market due to unknown reason, probably security issues. It was a very serious problem for my team. We tried to prepare the map from scratch going to evershop finding their shop number and shop name, steet nu etc. We went every shop and drew street map manually with paper and pencil. It was very hard for my team. I was very upset on the municipality manager about this. We were also frustrated very much because we did not have much time to prepare the digital map. At the end we managed to make a working map.
- I learned a professional value in solving problems and cooperate with my colleague to reach an agreement and to sub-divide the tasks to be able to finish them ahead of time although there were some difficulties. its value will give me an advantage where I can

FIGURE 9. A sample image of the reflection of a developer.


From: [redacted] <m[redacted]@student.qu.edu.qa>
Date: Tuesday, 2 Nov 2019 at 4:28 PM
To: Khaled Khan <k.khan@qu.edu.qa>
Subject: CMPS310 Project Milestone 2 - Team G03

Dear Dr Khaled,
Sorry that we missed the dealine by one day to submit Milestone-2 of the SW Engineering Project.
I am working with team G03.
I like to mention you that I could not concentrate much with my tasks of developing sequence diagrams of 6 use cases.. I felt gloomy,sad and bored. I cannot think properly last few days. There is a personal issue. However, we managed to submit it by one day late. I hope you will accept it .Thank you sir for consideration.

Emotion: Pensiveness (gloomy, sad, boredom)
Affected artifact: all sequence diagrams (2nd milestone)
Checked: SDs are good. A few consistency issues with the class diagram

[redacted]vari
SID [redacted]

FIGURE 10. A sample image of email of a developer.

developer contributed positively. The source of the anger was an external organization (a stakeholder) that had promised to provide her with a digital map of a traditional, old marketplace. She was a member of the team who was responsible for developing the software for this market. Later, the organization declined to provide the digital map in the middle of the project. It was so devastating that she became angry, and this anger later turned into resilience that also influenced other team members by empathy. The team was so confident that they created the required detailed digital map from scratch. Figure 9 depicts an image of her reflection in the final project report.

Another developer was suffering from pensiveness about her skills. She expressed this feeling in various communication artifacts. However, she later managed to control her emotion, ultimately turned this negative emotion into a positive one, and contributed to better quality artifacts. These findings confirm that sometimes an emotion, regardless of whether it is negative or positive, may contribute to either good or bad software artifacts. Figure 10 shows the email communication of the developer with the project supervisor. The name of the developer has been suppressed.

Table 11 shows the summary of the comparison of our results with other similar research work. Our study also shows that emotions impacting on software products are associated with one or more causes which can be external to the software engineering process or even personal. Supervisors have little to do in tackling developers' emotions caused by factors outside of the project. Here are some identified sources of expressed emotions:

- *Competition*: the post-development interviews showed that the team developers were competing with each other to produce good quality artifacts. This competition ultimately led to the distraction of one developer.
- *Challenges*: anger motivated a developer to strive hard to overcome the obstacles of acquiring a digital map required for her project. She replaced her anger with resilience to tackle the challenge.
- *Demotivation*: the communication artifacts revealed that the demotivation of two developers led to their boredom in the project. They disclosed that they were demotivated because they did not have the required expertise for the project.
- *Inferior and Superior Complexity*: two developers suffered from distraction due to the feeling of inferior complexity. One was more engaged in this complexity without trying to address her deficiencies. The other

developer was bored because she thought being inferior compared to other developers in his team. However, she later overcame her deficiencies, and that resulted in a positive impact. One developer of another team experienced superior complexity due to her over-confidence and played dominating roles. This later contributed to an inferior quality artifact due to the non-cooperating attitude of other team members.

- *Personal*: some developers experienced joy as well as anger due to a personal situation. It was revealed that the personal relationships, health issues, the financial situation of the developers are the sources for various emotions.
- *Work Stress*: due to work stress, a few developers experienced negative emotions like distraction and boredom. They reported that they were 'burnout' due to the intense required deliverables with firm deadline of their projects.
- *Intra-team Relationship*: The developers' happiness and confidence contributed to maximizing the complementary expertise to each other of a team and developed a bond among team members. They felt a sense of belonging to the project. This observation confirms the study reported in [39].

However, this study has some limitations. The number of participants in this research is twenty. At first glance, it looks insufficient for a better evaluation of the results. Note that this study is not based on a mere one-off survey; rather it followed the entire life cycle of the development process of five projects, captured three milestones deliverables of each team, track the quality of those delivered artifacts, and conducted interviews with the individual participants with a questionnaire, and used manual walk-through of artifacts. It has been a lengthy duration. Another issue is that Qatar is relatively a tiny country with fewer qualified software engineers and computer science graduates. Despite this, we acknowledge the fact that more participants would make this study more better evaluation. We acknowledge that the measurement of emotion is difficult in this type of study because our plan to measure emotions often does not permit us to observe the emotion of software developers in real-time during the project activities.

VIII. CONCLUSION

The paper has reported the role of different types of emotion in the software development process and their impacts on the quality of software artifacts. We have adapted Plutchik's wheel of emotions to examine the effect of emotion in the software development process. We have demonstrated how negative and positive emotions can have different effects on the quality of software products. Expressed and suppressed emotions of software developers have causes that could be related to factors not always associated with their workplace.

The results show that in some cases, negative emotion can produce better quality artifacts. A carefully catered development environment could reduce negative emotions and even encourage developers to turn their negative emotions into a positive impact on the quality of software products. Moreover, organizations could influence the emotions of developers by inducing different incentives. However, there is a limit to this approach. A well-defined emotional awareness can minimize negative emotions by introducing incentives, thus eliminating the negative impact on the quality of software products.

There are a few specific areas that further research could be conducted. Currently, the proposed Emotion-Landscape framework (Figure 8) does not directly represent any data automatically. Attempts could be initiated to make this landscape dynamic, meaning the attributes of the diagrams, such as the intensity of various colors and the space of the rectangles and the diagonal for negative-positive emotions, will automatically change based on the data fed. In other words, these attributes will readjust themselves automatically according to the data integrated into the diagram. A dynamic emotion-landscape diagram can capture the impact of emotions of a software development project with a summary map.

Another possible research front could be applying artificial intelligence to build correlations between the quality of artifacts and the emotions mined from the communication artifacts and interviews. To achieve this, various coded annotations can be used to represent quality variations of each software artifact component and intensity of emotions found in communication artifacts. AI-based algorithms could establish and reason about correlations between software artifact quality and emotions on a continuous scale as opposed to discrete binary values.

ACKNOWLEDGMENT

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) through the National Priorities Research Program (NPRP) under Grant NPRP-09-079-1-013. The findings achieved herein are solely the responsibility of the authors. Open Access funding is provided by the Qatar National Library.

REFERENCES

[1] M. R. Wrobel, "Emotions in the software development process," in *Proc. 6th Int. Conf. Human Syst. Interact. (HSI)*, Jun. 2013, pp. 518–523.

[2] T. Shaw, "The emotions of systems developers: An empirical study of affective events theory," in *Proc. Conf. Comput. Personnel Res. Careers, Culture, Ethics Netw. Environ. (SIGMIS CPR)*, 2004, pp. 124–126.

[3] D. Graziotin, X. Wang, and P. Abrahamsson, "Are happy developers more productive?" in *Proc. 14th Int. Conf. (PROFES)*, 2013, pp. 50–64.

[4] D. Kim, M. G. Frank, and S. T. Kim, "Emotional display behavior in different forms of computer mediated communication," *Comput. Hum. Behav.*, vol. 30, pp. 222–229, Jan. 2014.

[5] D. Graziotin, X. Wang, and P. Abrahamsson, "Software developers, moods, emotions, and performance," *IEEE Softw.*, vol. 31, no. 4, pp. 24–27, Jul. 2014.

[6] S.-H. Park, B.-C. Bae, and Y.-G. Cheong, "Emotion recognition from text stories using an emotion embedding model," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2020, pp. 579–583.

[7] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.

[8] R. Plutchik, "The nature of emotions: Clinical implications," in *Emotions and Psychopathology*, M. Clynes and J. Panksepp, Eds. Boston, MA, USA: Springer, 1988.

[9] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: The PANAS scales," *J. Personality Social Psychol.*, vol. 54, no. 6, pp. 1063–1070, 1988.

[10] Y. Rao, J. Lei, L. Wenyin, Q. Li, and M. Chen, "Building emotional dictionary for sentiment analysis of online news," *World Wide Web-Internet Web Inf. Syst.*, vol. 17, no. 4, pp. 723–742, 2014.

[11] Y. Rao, Q. Li, X. Mao, and L. Wenyin, "Sentiment topic models for social emotion mining," *Inf. Sci.*, vol. 266, pp. 90–100, May 2014.

[12] Y. Rao, Q. Li, L. Wenyin, Q. Wu, and X. Quan, "Affective topic model for social emotion detection," *Neural Netw.*, vol. 58, pp. 29–37, Oct. 2014.

[13] Q. Yang, Y. Rao, H. Xie, J. Wang, F. L. Wang, and W. H. Chan, "Segment-level joint topic-sentiment model for online review analysis," *IEEE Intell. Syst.*, vol. 34, no. 1, pp. 43–50, Jan./Feb. 2019.

[14] M. Dragoni, S. Poria, and E. Cambria, "OntoSenticNet: A commonsense ontology for sentiment analysis," *IEEE Intell. Syst.*, vol. 33, no. 3, pp. 77–85, May 2018.

[15] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1795–1802.

[16] A. Esuli and F. Sebastiani, "SentiWordNet: A publicly available lexical resource for opinion mining," in *Proc. 5th Int. Conf. Lang. Resour. Eval. (LREC), Eur. Lang. Resour. Assoc. (ELRA)*, 2006, pp. 417–422.

[17] M. R. Islam and M. F. Zibran, "Leveraging automated sentiment analysis in software engineering," in *Proc. IEEE/ACM 14th Int. Conf. Mining Softw. Repositories (MSR)*, May 2017, pp. 203–214.

[18] N. Novielli, F. Calefato, and F. Lanubile, "The challenges of sentiment detection in the social programmer ecosystem," in *Proc. 7th Int. Workshop Social Softw. Eng.*, Sep. 2015, pp. 33–40.

[19] X. Li, Y. Rao, H. Xie, R. Y. K. Lau, J. Yin, and F. L. Wang, "Bootstrapping social emotion classification with semantically rich hybrid neural networks," *IEEE Trans. Affective Comput.*, vol. 8, no. 4, pp. 428–442, Oct./Dec. 2017.

[20] S. F. Huq, A. Z. Sadiq, and K. Sakib, "Is developer sentiment related to software bugs: An exploratory study on GitHub commits," in *Proc. IEEE 27th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Feb. 2020, pp. 527–531.

[21] J. O'Dwyer, "Speech, head, and eye-based cues for continuous affect prediction," in *Proc. 8th Int. Conf. Affect. Comput. Intell. Interact. Workshops Demos (ACIIW)*, Cambridge, U.K., Sep. 2019, pp. 16–20.

[22] L. He, D. Jiang, L. Yang, E. Pei, P. Wu, and H. Sahli, "Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks," in *Proc. 5th Int. Workshop Audio/Vis. Emotion Challenge*, Oct. 2015, pp. 73–80.

[23] F. Ringeval, F. Eyben, E. Kroupi, A. Yuce, J.-P. Thiran, T. Ebrahimi, D. Lalanne, and B. Schuller, "Prediction of asynchronous dimensional emotion ratings from audiovisual and physiological data," *Pattern Recognit. Lett.*, vol. 66, pp. 22–30, Nov. 2015.

[24] S. Tripathi, S. Acharya, R. D. Sharma, S. Mittal, and S. Bhattacharya, "Using deep and convolutional neural networks for accurate emotion classification on DEAP dataset," in *Proc. IAAI*, 2017, pp. 4746–4752.

[25] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. ACL Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2002, pp. 79–86.

[26] J. Pang, Y. Rao, H. Xie, X. Wang, F. L. Wang, T.-L. Wong, and Q. Li, "Fast supervised topic models for short text emotion detection," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 815–828, Feb. 2021.

[27] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 496–509, Feb. 2016.

[28] J. Li, Y. Rao, F. Jin, H. Chen, and X. Xiang, "Multi-label maximum entropy model for social emotion classification over short text," *Neurocomputing*, vol. 210, pp. 247–256, Oct. 2016.

[29] A. Chatchinarat, K. W. Wong, and C. C. Fung, "Fuzzy classification of human emotions using fuzzy C-mean (FCFCM)," in *Proc. Int. Conf. Fuzzy Theory Appl. (iFuzzy)*, Taichung, Republic of China, Nov. 2016, pp. 1–5.

[30] D. Martens and W. Maalej, "Release early, release often, and watch your users' emotions: Lessons from emotional patterns," *IEEE Softw.*, vol. 36, no. 5, pp. 32–37, Sep./Oct. 2019.

[31] R. Maipradit, H. Hata, and K. Matsumoto, "Sentiment classification using N-gram inverse document frequency and automated machine learning," *IEEE Softw.*, vol. 36, no. 5, pp. 65–70, Sep./Oct. 2019.

[32] S. Koelstra, C. Muhl, M. Soleymani, J. Lee, A. Yasdani, and T. Ebrahimi, "DEAP: A database for emotion analysis; using physiological signals," *IEEE Trans. Affective Comput.*, vol. 3, no. 1, pp. 18–31, Oct./Mar. 2012.

[33] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, "A multimodal database for affect recognition and implicit tagging," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 42–55, Jan. 2012.

[34] L. Shu, J. Xie, M. Yang, Z. Li, Z. Li, D. Liao, X. Xu, and X. Yang, "A review of emotion recognition using physiological signals," *Sensors*, vol. 7, pp. 1–42, Jul. 2018.

[35] S. Vijayakumar, R. Flynn, and N. Murray, "A comparative study of machine learning techniques for emotion recognition from peripheral physiological signals," in *Proc. 31st Irish Signals Syst. Conf. (ISSC)*, Jun. 2020, pp. 1–6.

[36] D. Girardi, F. Lanubile, N. Novielli, L. Quaranta, and A. Serebrenik, "Towards recognizing the emotions of developers using biometrics: The design of a field study," in *Proc. IEEE/ACM 4th Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, May 2019, pp. 13–16.

[37] A. Murgia, P. Tourani, B. Adams, and M. Ortu, "Do developers feel emotions? An exploratory analysis of emotions in software artifacts," in *Proc. 11th Work. Conf. Mining Softw. Repositories (MSR)*, vol. 14, Jun. 2014, pp. 262–271.

[38] A. Fountaine and B. Sharif, "Emotional awareness in software development: Theory and measurement," in *Proc. IEEE/ACM 2nd Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, May 2017, pp. 28–31.

[39] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "What happens when software developers are (un) happy," *J. Syst. Softw.*, vol. 140, pp. 32–47, Jun. 2018.

[40] A. Serebrenik, "Emotional labor of software engineering," in *Proc. BENEVOL*, 2017, pp. 1–6.

[41] A. Gunsel, "The effects of emotional labor on software quality: The moderating role of project complexity," *J. Global Strategic Manage.*, vol. 8, no. 2, pp. 101–115, 2014.

[42] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, and M. Ortu, "Mining valence, arousal, and dominance: Possibilities for detecting burnout and productivity?" in *Proc. 13th Int. Conf. Mining Softw. Repositories*, May 2016, pp. 247–258.

[43] M. R. Wrobel, "The impact of lexicon adaptation on the emotion mining from software engineering artifacts," *IEEE Access*, vol. 8, pp. 48742–48751, 2020.

[44] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Softw. Eng.*, vol. 23, no. 3, pp. 1352–1382, Jun. 2018.

[45] L. Clark and D. Watson, "Distinguishing functional from dysfunctional affective responses," in *The Nature of Emotion: Fundamental Questions*, P. E. Ekman and R. J. Davidson, Eds. Oxford, U.K.: Oxford Univ. Press, 1994.

[46] S.-B. Kim, K.-S. Han, H.-C. Rim, and S. H. Myaeng, "Some effective techniques for naive Bayes text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 11, pp. 1457–1466, Sep. 2006.

[47] N. Novielli and C. Strapparava, "The role of affect analysis in dialogue act identification," *IEEE Trans. Affect. Comput.*, vol. 4, no. 4, pp. 439–451, Oct. 2013.

[48] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive? Empirical study of affectiveness vs. issue fixing time," in *Proc. IEEE/ACM 12th Work. Conf. Mining Softw. Repositories*, Florence, Italy, May 2015, pp. 303–313.

[49] C. Ringle, S. Wende, and A. Will. (2005). *SmartPLS 3.0 (M3) Beta*. [Online]. Available: http://www.smartpls.de

[50] C. M. Ringle and R. R. Sinkovics, "The use of partial least squares path modeling in international marketing," *Adv. Int. Marketing*, vol. 20, no. 1, pp. 277–319, Mar. 2009.

[51] J. Hair, M. Sarstedt, C. Ringle, and J. Mena, "A. An assessment of the use of partial least squares structural equation modeling in marketing research," *J. Acad. Marketing Sci.*, vol. 40, no. 3, pp. 414–433, 2021.

[52] W. Chin, "How to write up and report PLS analyses," in *Handbook of Partial Least Squares: Concepts, Methods and Applications in Marketing and Related Fields*, V. E. Vinzi, W. W. Chin, J. Henseler, and H. Wang, Eds. Berlin, Germany: Springer, 2010, pp. 655–690.

[53] C. Fornell and D. F. Larcker, "Evaluating structural equation models with unobservable variables and measurement error," *J. Marketing Res.*, vol. 18, no. 1, pp. 39–50, Feb. 1981.

[54] D. Szczygieł, J. Buczny, and R. Bazińska, "Emotion regulation and emotional information processing: The moderating effect of emotional awareness," *Personality Individual Differences*, vol. 52, no. 3, pp. 433–437, Feb. 2012.

[55] P. S. Rutner, F. I. Williams, C. Campbell, and C. K. Riemenschneider, "The politics of emotion: Exploring emotional labor and political skill across job types within the IT/IS profession," *ACM SIGMIS Database, DATABASE Adv. Inf. Syst.*, vol. 46, no. 3, pp. 52–73, Jul. 2015.
[56] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "Consequences of unhappiness while developing software," in *Proc. IEEE/ACM 2nd Int. Workshop Emotion Awareness Softw. Eng. (SEmotion)*, May 2017, pp. 42–47.
[57] D. Graziotin, X. Wang, and P. Abrahamsson, "Happy software developers solve problems better: Psychological measurements in empirical software engineering," *PeerJ*, vol. 2, no. 1, p. e289, Mar. 2014.
[58] I. A. Khan, W.-P. Brinkman, and R. M. Hierons, "Do moods affect programmers' debug performance?" *Cognition, Technol. Work*, vol. 13, no. 4, pp. 245–258, Nov. 2011.

KHALED M. KHAN (Senior Member, IEEE) received the B.S. and M.S. degrees from the Norwegian University of Science and Technology, and the Ph.D. degree from Monash University. He was with Western Sydney University, Australia, and the Head of Graduate Programs. He is currently an Associate Professor of computer science and engineering with Qatar University.

MOUTAZ SALEH received the B.Sc. degree in computer engineering from Yarmouk University, Jordan, the M.Sc. degree in distributed computing from Universiti Putra Malaysia, and the Ph.D. degree in computer science from Universiti Kebangsaan Malaysia. He worked as a Cisco Certified Network Engineer with Jeraisy Computer and Communication Services (JCCS), Riyadh, Saudi Arabia, and MobileCom Telecommunications, Amman, Jordan, from 2001 to 2003. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Qatar University. He is also a Cisco Certified Internetwork Expert (CCIE) and a Cisco Certified Instructor Academy (CCAI) with Qatar University Cisco Academy. He has published over 50 refereed articles in reputable international journals and conferences. His research interests include text-to-image synthesis, assistive technology, educational multimedia systems, natural language processing, software engineering, network scheduling, and distributed and real-time systems. He received several awards and collaborated on research projects funded by Qatar National Research Fund. He has served as a Technical Program Committee for *Computer Communications* (Elsevier), *EURASIP Journal on Wireless Communications and Networking*, and IEEE ACCESS, IEEE LCN, IEEE ICT, IEEE AICCSA, IEEE APCC, IEEE GCC, IEEE ICC, IEEE EDUCON, and IEEE TALE.