The Journal of Systems and Software 159 (2020) 110434

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Affect Recognition in Code Review: An In-situ Biometric Study of Reviewer's Affect

Hana Vrzakova[a,*], Andrew Begel[b], Lauri Mehtätalo[a], Roman Bednarik[a]

[a] University of Eastern Finland, School of Computing, Joensuu, Finland
[b] Microsoft Research, Redmond, Washington, U.S.A.

ARTICLE INFO

Article history:
Received 10 December 2018
Revised 16 September 2019
Accepted 8 October 2019
Available online 8 October 2019

Keywords:
Code Review
Affective Computing
Physiological Signals
CSCW

ABSTRACT

Code reviews are an important practice in software development that increases team productivity and improves product quality. They are also examples of remote, computer-mediated asynchronous communications which are prone to the loss of affective information. Prior research has focused on sentiment analysis in source codes, as positive affect has been linked to developer productivity. Although methods of sentiment analysis have advanced, challenges remain due to numerous domain-specific expressions, subtle nuance, and indications of sentiment. In this paper, we uncover the potential for 1) nonverbal behavioral signals such as conventional typing, and 2) indirect physiological measures (eye gaze, GSR, touch pressure) to reveal genuine affective states in *in situ* code review in a large software company.

Nonverbal behavioral signals of 33 professional software developers were recorded unobtrusively while they worked on their daily code reviews. After analyzing these signals using Linear Mixed Effect Models, we observe that affect presented in the written comments is associated with prolonged typing duration. Using physiological features, a trained Random Forest classifier can predict post-task valence with 90.0% accuracy (F1-score = 0.937) and arousal with 83.9% accuracy (F1-score = 0.856). The results show promise for the creation of intelligent affect-aware interfaces for code review.

1. Introduction

The ability to effectively communicate and interpret affect and emotions is central to human daily activity, and is considered one of the key skills for functional team collaboration (Islam and Zibran, 2018; Schneider et al., 2018; Graziotin et al., 2018). In remote, computer-mediated interaction and textual communication, however, social-behavioral signals, especially non-verbal behaviors, are attenuated (Hancock et al., 2007; Schulze and Krumm, 2017), and consequently, affective information becomes less salient or is even lost.

In these indirect computer-mediated contexts, interpretation of affective states is very hard (Riordan and Trichtinger, 2017) because textual representations contain less affect than a phone call (for example) (Picard, 1999). Consequently, the tone of the message can quickly, unnoticeably, and yet significantly, change affective polarity; humor may be interpreted as offense, a critique may sound inadequate and harsh, serious messages may be ignored, and constructive notes may suddenly appear to be jokes. In the software development industry, affect processing, understanding, and communication have been central to the success of development processes. Understandably, positive affect in intra- and inter-team communication is beneficial for performance and productivity (Wrobel, 2013; Müller and Fritz, 2015; Schneider et al., 2018; Graziotin et al., 2018).

The loss of affect becomes more important in large-scale distributed teams, especially as the number of spatially-remote teams increases (Herbsleb and Mockus, 2003). Consequently, software businesses are responding through increased use of collaborative virtual environments and tools (Storey et al., 2017). These tools, however, lack the ability to support effective affect-signalling and recognition. In this work, we focus on developing methods and tools that can enhance the communication channels through automatic recognition of a code reviewer's affective state.

Source code review is an example of a software development activity that evolved from a human-to-human interaction at arranged meetings to asynchronous and remote computer-mediated textual communications interactions (Bacchelli and Bird, 2013). Although saving time and speeding up product release cycles, the contemporary form of code review is arguably prone to affective loss. To extract the sentiment and emotions in written text, prior

* Corresponding author. Tel.: +358406501417.
E-mail addresses: hana.vrzakova@uef.fi (H. Vrzakova), andrew.begel@microsoft.com (A. Begel), lauri.mehtatalo@uef.fi (L. Mehtätalo), roman.bednarik@uef.fi (R. Bednarik).

https://doi.org/10.1016/j.jss.2019.110434
0164-1212/

research has employed numerous machine-learning tools for sentiment detection (for review, see e.g. Tang et al. (2009); Mäntylä et al. (2018)).

In software development, however, communication between developers is often centered around the project tasks and obstacles, and populated by snippets of code specific to the project's programming language. Since the tools for sentiment analysis struggle with this variety, current research has been developing software engineering-specific detection algorithms such as SentiStrenghtSE (Islam and Zibran, 2018), Senti4SD (Calefato et al., 2018), and SentiCR (Ahmed et al., 2017). Note, SentiCR is specific to code review. However, aspects such as context-sensitive variations of words, subtle expressions of sentiment, humor, irony, or sarcasm, politeness, or missing explicit polarized lexical cues can still hinder effective recognition of affect (Islam and Zibran, 2018; Novielli et al., 2018).

In this work, we build and evaluate a multimodal recognition of affect from programmers' nonverbal signals during the code review tasks. While current research on affect recognition has employed directly observable modalities to recognize basic and elicited emotions (predominantly from facial and acoustic-prosodic expressions, for review see e.g. D'mello and Kory (2015)), the context of remote, asynchronous code review is less suited for such a approaches, due to the lack of speech activity as well as privacy considerations.

Our work differentiates and advances the prior research in numerous aspects and also presents several novel contributions. We collected both behavioral and physiological signals, namely typing behavior, eye gaze, galvanic skin response, and touch pressure, during the *in situ* code review in a very large software company.

Across a variety of code review tasks, we model and analyze *unelicited* affective states from multiple perspectives, a goal towards automatic recognition of reviewer's affect. We approach affect recognition from three perspectives. First, we analyze effects of long-term positive and negative affect (*i.e.* the mood of the reviewer) as well as task-related aspects and how they influence the reviewer's affect after the task. Next, we analyze commenting behavior metrics (typing duration and comment length) for each participant in relation to emotions in each of their individual comments; we perform this analysis since such approach does not require any dedicated sensors. Third, in the analysis of each participant's affect, we extract features related to physiological states and employ a machine-learning based framework to distinguish valence and arousal polarity.

In sum, we center our research on the following questions:

1. *How do long-term affect and task related aspects predict the components of affect after the code-review task?*
2. *How does the presence of emotion influence commenting behavior, i.e. typing on a keyboard and comment length?*
3. *How do nonverbal physiological signals predict components of affect after the code review task?*

Effective communication of affect has been gaining traction in the software engineering domain. In this work we weigh on the benefits of nonverbal multimodal approaches and their potential for future *affect-enhanced* code review.

The rest of paper is organized as follows. Section 2 introduces the domain of code review, provides an overview of the study of physiological signals in software engineering. In Section 3, we present experimental settings of the *in situ* study; we propose two analysis methods in Section 4. Section 5 summarizes the results of the analyses using comment-related behavioral measures and the results of the machine learning analysis using physiological signals. In Section 6, we discuss the results in the light of current research, limitations, and future directions of affect recognition in code review.

2. Background

The task of affect recognition in code review spans the fields of software engineering, affective computing, computer-supported collaborative work, and inference of user states from physiological signals. We provide a brief introduction to a range of corresponding studies with respect to software engineering. We report on a first of its kind, multimodal investigation of affect in code review that was performed *in situ*; we recorded the data in one of the world's largest software companies with professional code reviewers engaged in their everyday tasks.

2.1. Origins of code review

Contemporary code review practices originated in *code inspections,* in which source code was scrutinized at formal project group assemblies that sometimes last days (Fagan, 1999). Due to the fast pace of software production and the understandable impracticality of the long face-to-face meetings, code inspection became a computer-mediated task through dedicated interfaces and tools, such as CodeFlow (see Figure 1). In principle, code review still resembles a code inspection session where a developer evaluates another's code, looking for potential errors, and suggesting improvements. However, the current form has evolved into an informal, lightweight, and brief code review practice.

The evolution of code review practice came along with the development of tools dedicated for code review. To an external observer, current user interfaces are indistinguishable from fully functional IDEs with access to linked libraries, classes, and relevant resources. Reviewing code is no different than commenting a shared document where comments are directly linked to the relevant piece of code. In addition, all the comments are synchronized and shared between reviewers who are invited to that particular code review.

Code reviewers use various strategies (Peng et al., 2016; Uwano et al., 2006). Some reviewers briefly proofread the code, check on code style, and search for obvious typo-like errors. Others look for logic errors, and provide mentoring in their comments (Bacchelli and Bird, 2013; Ebert et al., 2018). Similar to revisions in writing, code review comments are addressed in iterations, which helps to direct a reviewer's strategy for proofreading. While the first rounds of code review may require focused reading, later reviews of the same code may be quick, simply scrutinizing whether the reviewer's recommendations were implemented.

Independent of the selected strategy, the reviewer's reasoning, decision-making rationale, and affective states remain hidden to the author of the code unless explicitly written in the comments. Furthermore, code review comments are often brief, factual, and free of emotions. This conforms with common, tacit, engineering-focused cultures whose primary goal is to get work done, not antagonize one's colleagues (Lutchyn et al., 2015). Therefore, in code review, affective information gets lost due to the form of communication, computer-mediation, and professional conduct. Understanding affect in such settings is crucial for efficient functioning in teams (De Choudhury and Counts, 2013; Dewan, 2015), and to contribute to effective communication and coordination (Herbsleb et al., 1995; Schneider et al., 2018).

2.2. Affect in software engineering

Understanding the developer's affect and their team's emotional awareness, in particular, underlies effective software engineering (Dewan, 2015). To general audiences, however, the domain of software engineering seems to lack extreme and overt affective expressions. Software developers often appear calm, focused, or distant. On the contrary, since software development is highly depen-

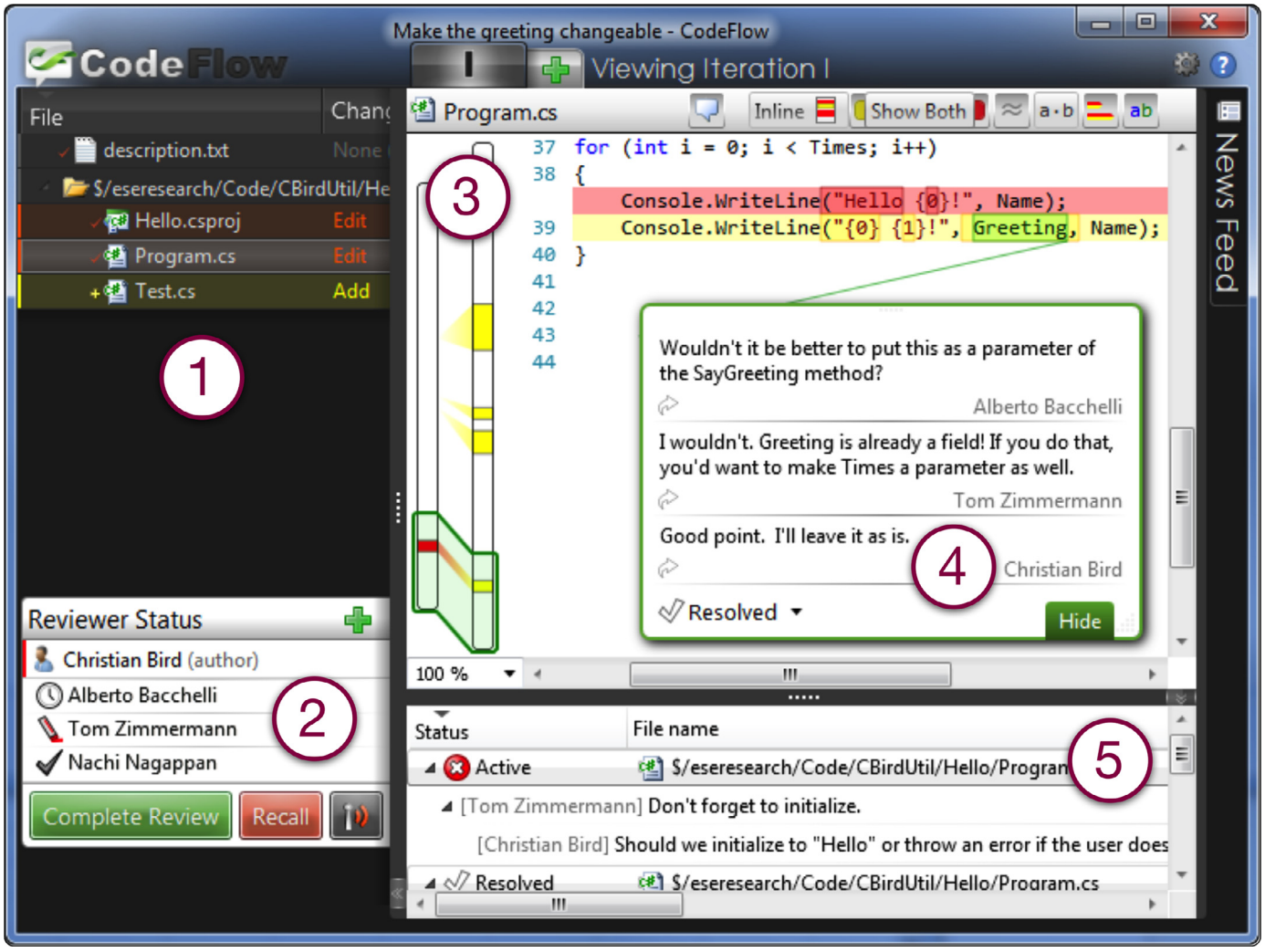

Fig. 1. A typical user interface of a corporate code review tool (CodeFlow). A source code file is selected in a file listing (1), displayed in the main window (3), and commented in pop-up windows (4). A review summary along with other people's comments are displayed below the code (5). Adopted from Bacchelli and Bird (2013).

dent on developers' cognitive efforts, their performance, productivity, and creativity must be influenced by affect (Islam and Zibran, 2018; Schneider et al., 2018). Happiness among programmers has been found to correlation with productivity (Graziotin et al., 2013; 2018), while negative emotions are often detrimental for the software development (Müller and Fritz, 2015; Gachechiladze et al., 2017) (though not exclusively (Wrobel, 2013)).

Affect assessment during software development is challenging (Schmidt et al., 2018). One explanation could be that when one is engaged in already cognitively-demanding task, such as code review, reflecting on one's cognitive processing and states becomes tremendously difficulty and elevates a participant's overall high workload (Ericsson and Simon, 1993). Therefore, traditional qualitative methods of affect assessment fall short and may be perceived to be costly, time demanding, and challenging for adoption in software development (Schmidt et al., 2018; Lutchyn et al., 2015).

Prior analyses of developer's internal states, for example, of happiness (Graziotin et al., 2013; 2018), frustration (Hernandez et al., 2014; Müller and Fritz, 2015), anger (Gachechiladze et al., 2017), stress (Sano et al., 2017; Sarker et al., 2019), and workload (Fritz et al., 2014), have laid the foundation for the development of novel inferential methods such as sentiment analysis from source code resources (Islam and Zibran, 2018) and non-verbal sensing from direct and indirect physiology (D'mello and Kory, 2015; Shu et al., 2018).

To estimate increased workload during an SE task, Fritz et al. Fritz et al. (2014) employed galvanic skin response (GSR) and electroencephalography (EEG) together with eye tracking sensors. In their analysis of multimodal signals, a programmer's perceived difficulty with the code was predictable with 84% precision on every new task. To measure frustration during daily work, Hernandez et al. Hernandez et al. (2014) averaged signals from a pressure-sensitive keyboard and a capacitive mouse (Microsoft TouchMouse). Under stressful conditions, both typing pressure and contact with the mouse increased in 75% of participants. In the context of software change tasks, Müller et al. Müller and Fritz (2015) measured developers' valence and feeling of progress using GSR, EEG, and heart rate. In their analysis of valence, a classifier was able to predict a developer's emotional reaction with 71.36% accuracy and their feeling of progress with 67.70% accuracy. Signals of indirect physiology and user's activity have also been employed in recognition of stress and suitable timing of stress micro-interventions. Sano et al. Sano et al. (2017) used an array of sensors to sense a developer's activity, heart rate variability (HRV), and intervention history. These sensors were used to predict the time when a preventive intervention should be delivered to be both efficient and unobtrusive. Multi-kernel SVM techniques could

be used to differentiate suitable and unsuitable timing with 80% accuracy.

A drawback of the previous, highly controlled lab-based studies with biometric sensors comes from the characteristics of the source code presented. The code used in the experiments has been usually isolated, shortened, or simplified to fit the screen. In addition, materials were not connected to the routine work of the participants, nor related to their projects and responsibilities. In addition, the participants had little to no means of interacting with the code, such as scrolling, opening files, switching to necessary libraries, and searching for supporting code. In a real world scenario, however, a single source code can easily cover hundreds of lines, is a part of a larger package or a project, and is created and maintained by numerous programmers in the team. Understandably, due to the demanding setup of physiological sensing, prior *in situ* studies of affect recognition in software development have centered on self-assessed psychological measurements during and after the task Graziotin et al. (2015); Kuutila et al. (2018).

In this work, we investigate overt behaviors (typing behavior) and physiological signals (eye movements, GSR, and touch) that have not been extensively studied before, but which can, however, be embedded into the daily work environments. We utilize three sensors of measuring indirect physiology with great potential for ubiquitous sensing computers, and we link the biometric signals to each code reviewer's self-assessed affective state.

3. Experiment in-situ

In our study, we observed professional software developers conducting remote asynchronous code reviews as part of their daily work at a large international software company. We instrumented participants with three wearable sensors: a Shimmer GSR to measure electrodermal activity (EDA), a 195-point Microsoft capacitive TouchMouse to measure stress levels, and a portable remote Tobii eye tracker to identify reviewer's focus in the code. We chose affordable and mobile sensors that are simple to embed in near-future computers and that do not require lengthy instrumentation of users. All three sensors have been previously validated in multiple studies as reliable and accurate for recording physiological signals (Burns et al., 2010; Hernandez et al., 2014; Coyne and Sibley, 2016; Huang et al., 2016; Gibaldi et al., 2017).

3.1. Task and procedure

The experiment was designed as an *in situ* study held in each participant's office. We brought all the experimental equipment for each session and installed it into each participant's environment. After the participants became familiar with the setup of the biometric sensors and signed a consent form, they answered a set of preparatory questionnaires related to their experience with the code review process in the company. We then employed the PANAS measurement tool (Watson et al., 1988) to assess their long-term affective state.

As participants answered the questionnaires, we installed the sensors on the experimental computer, plugged it into the participant's primary monitor, and mirrored the participant's work account. The account mirroring and minimal changes in participant's environment ensured high levels of experimental ecological validity. Finally, we calibrated the eye-tracker using a 9-point calibration method and validated the calibration by asking the participant to read aloud the first and the last visible line of the source code at the screen.

After the 15 minute calibration stage, the participants were free to open up the code review of their choice and work on their task in CodeFlow as long as needed (all finished in under 45 minutes). After each task, participants were asked to assess their affective state about the code, the author of the code, and themselves (using the PAM scale (Pollak et al., 2011)). Participants also reported their familiarity with the reviewed code, binned into three levels (low, medium, or high). Similarly, participants specified their work-related organizational hierarchy towards the author of the code under review (work seniority lower, equal, or higher). Finally, task difficulty was evaluated using using the NASA TLX instrument (Hart and Staveland, 1988). Table 1 summarizes the collected assessments.

At the end of the experimental session, the participant was offered a remuneration of an $8 company cafeteria coupon. Recordings from four participants were excluded from the further analysis due to malfunctioning of the recording setup during the experiment (n=2), missing PAM outcomes (n=1), or unexceptionally short duration of the review (n=1).

3.2. Participants

We examined biometrics signals of 37 software developers (2 female, 35 male). The age of the participants ranged from 25 to 43 years (mean = 34 years, SD = 4.74). Each participant was the member of a team responsible for building and shipping consumer-focused software products and/or services. Potential participants were identified through a search of the company-wide code review database. We invited anyone who was an active reviewer, had completed more than ten reviews in the past three months, and had more than five code reviews pending. Our selection criteria ensured that our study participants had sufficient experience with the CodeFlow interface, as opposed to interns and newly hired developers. Since all code reviewers were assigned to new code review daily, we repeated our search and emailed potential candidates whether they would be willing to participate in the experiment every day.

3.3. Tools and apparatus

Preserving the *in situ* nature of the study was important for mimicking future interactions with the the company's code review environment. We instrumented participants with three portable biometric sensors: Tobii EyeX[1] (60Hz, binocularly), Shimmer3 GSR+ electrodermal activity[2] (EDA) sensor, and 195-point Microsoft TouchMouse Ultimate[3].

All sensors were integrated into the code review environment using a corresponding API and synchronized with the reviewer's interaction over Bluetooth. In addition to the physiological signals, we recorded the mouse position in the CodeFlow window. Preprocessing of the recorded data streams was performed using custom-made Python scripts with included libraries Scikit-learn (Pedregosa et al., 2011), Numpy (van der Walt et al., 2011), and Pandas(McKinney, 2010). Inferential analysis was performed using *lme4* and *nlme* libraries (Bates et al., 2015; Pinheiro et al., 2017) available in R (R Core Team, 2015). Training and evaluation of machine learning models was conducted at the Taito supercluster[4].

4. Analysis of reviewer's affect in comments

In this work, we evaluate how conventional metrics related to code review reveal affect occurrence in a review comment, and how nonverbal behavioral signals respond to long-term affect during the code review task. First, the conventional metrics were fitted to a Linear Mixed Effect model (LME) to see whether affect in-

[1] http://www.tobii.com/xperience/products/
[2] http://www.shimmersensing.com
[3] https://www.microsoft.com/accessories/en-gb/d/touch-mouse
[4] https://research.csc.fi/taito-supercluster

Table 1
Dependent and independent variables

Dependent variables	Description	Scale
Valence after the task		[1...4]
Arousal after the task	Photographic Affect Meter (Pollak et al., 2011)	[1...4]
Independent variables	**Description**	**Scale**
Long-term positive affect	PANAS scale (Watson et al., 1988)	[10...50]
Long-term negative affect		[10...50]
Code familiarity	Reviewer's familiarity with the code: low (i.e. seen for the first time), medium (i.e. worked on this review before), high (i.e. nth iteration of the review)	[low, medium, high]
Reviewer's seniority	Reviewer's work hierarchy with respect to the author of the code: lower (i.e. the reviewer is an intern), equal (i.e. the reviewer is a teammate), higher (i.e. the reviewer is a project lead)	[lower, equal, higher]
Task difficulty	Total of Task Load Index (NASA TLX) (Hart and Staveland, 1988)	[6...120]

fluenced the typing speed of the reviewer. Second, the reviewer's physiological signals are encoded to physiological features and employed in training of a Random Forest classifier to predict their valence and arousal after the task.

4.1. Affect in code review comments

All comments were first gathered from a CodeFlow database and annotated according to affect occurrence as *neutral* (negative class) or *emotional* (positive class). The authors of this paper annotated the data, as they are highly familiar with the task of code review (expert with code reviews in general, and with the particular company environment), and also familiar with the project culture. The annotation procedure was developed jointly. The exact annotation task was to determine the presence of affect-related words in each comment and the orientation of the comment — either objective neutral (towards the solution) or subjective affective opinion or expressions — i.e. accepting, patronizing, use of emoticons. During the annotation process, the two annotators first processed the 259 labels independently from each other, and then resolved the 84 disagreements in joint discussion. Though the initial inter-rater agreement was 66.92%, after joint discussion, they reached 100% agreement. Altogether, reviewers produced 259 comments: 238 neutral (92.31%) and 21 containing emotional content (7.69%). Comments with affect were produced by 10 reviewers (out of 33) who also wrote other neutral comments.

To investigate whether the affect in the comments can be captured by tools already available, we extracted metrics related to comments. Using database query, we exported the timestamps of comment start time and end time, along with the content of the comment. Each comment was represented by its *typing duration* (in seconds) and *comment length* (number of characters in the comment). We hypothesized that typing behavior is affected by the presence of emotion, when the comment length and variability in typing speed among reviewers are taken into account. Our initial hypothesis was that the speed of comment writing (comment length/comment duration) could be influenced by the preceding emotion. However, we decided it would be more natural to analyze the effects of emotionality on typing duration than on comment length.

We examined how *typing duration* was influenced by comment *emotionality* and *comment length* by fitting a Linear Mixed Effect model. We entered *comment emotionality* (EMO) as fixed effects and *comment length* (LEN) as main effects into the model. In addition, we hypothesized that the comment emotionality has an effect on typing speed. Therefore, we also included the interaction effect to remove the nuisance effect of comment length. It reveals whether emotional comments are written at a different speed than the non-emotional comments. As random effects, we had intercepts and random slopes for each *participant*. Nested random effects for *task* within a participant were also tried but omitted, since it explained a negligible amount of variance. Finally, we relaxed the assumption of constant residual variance, since the residuals were inclined to increase as a function of fitted value in the residual plots. Significance testing was obtained using Wald's F-tests of the full model. The resulting model is defined in Equation 4.1 with $var(\varepsilon_{ij})$ in Equation 2.

$$DUR_{ij} = \beta_0 + \beta_1 EMO_{ij} + \beta_2 LEN_{ij} + \beta_3 EMO_{ij} * LEN_{ij} + b_i^{(1)} + b_i^{(2)} LEN_{ij} + \varepsilon_{ij}\,, \quad (1)$$

In the model, DUR_{ij} denotes the typing duration for comment j of participant i, EMO_{ij} is the emotionality and LEN_{ij} is the centralized comment length (the original range was between 3 to 334 characters per a comment, mean = 77.02 characters, SD = 61.51), $(b_i^{(1)}, b_i^{(2)})'$ are the random effects for participant i, independent among participants and having bivariate normal distribution with mean zero and unknown variance, and ε_{ij} are independent normally distributed zero-mean residuals with variance

$$var(\varepsilon_{ij}) = \sigma^2 \left|\widehat{y}_{ij}\right|^{2\delta}. \quad (2)$$

During model development, visual inspection of residual plots and Q-Q plots revealed two outliers in the neutral class. In these two cases, the participants opened a comment and a web browser, and spent over four and seven minutes respectively typing and searching for additional information online. Since the comments did not present a typical behavior, we removed them.

4.2. Data preprocessing and feature engineering

Affordable biometric sensors often output noisy signals, which are unsuitable for direct statistical inference. To filter and clean the input data, we performed several transformations. Since data transfer during the experiment was established over Bluetooth, all data streams were recorded with a best effort sampling frequency and fluctuated in time. The frequency was unified to 50Hz using a mean of values and a backward propagation of missing values. If a data point was missing in the re-sampled data frame (a gap between data points was bigger than 20ms in the original data frame), the last data sample from the previous data frame was linearly approximated from the previous 20ms data segment. The resampling routine reestablished signal continuity.

Each data stream was filtered to remove noise. Raw GSR data was first normalized with Z-score and smoothed with an exponential filter ($\alpha = 0.08$). A decomposition of electrodermal activity followed the routine introduced by Fritz et al. (2014) splitting the signal into a *phasic component* (skin conductance response, SCR), which is associated with fast events as a shock or surprise, and a *tonic component* (skin conductance level, SCL), which responds to slow changes in autonomic arousal (Braithwaite et al., 2013). The SCL was extracted using low-pass Butterworth filter (0.05 Hz, 5^{th} order), revealing the slow trends in participant's arousal, while the SCR was obtained from a high-pass filter (0.33 Hz, 5^{th} order), capturing spikes in arousal.

Table 2
Features computed from indirect physiology.

Modality	Measure	Feature
GSR	Tonic component	Mean
		Median
		Variance
	Phasic component	Maximum
		Minimum
		Sum per s
Gaze	Euclidean distance	Mean
	Horizonal Euclidean distance	Median
	Vertical Euclidean distance	Variance
	Velocity	Maximum
	Horizontal velocity	Minimum
	Vertical velocity	Sum
TouchMouse	Sum of capacitive pixels	Mean
		Median
		Variance
	Number of fingers detected	Maximum
		Minimum
		Sum

Raw eye-tracking data was filtered in real-time during the experiment using a median filter with 10s sliding window to reduce the amount of missing data. To characterize attentional behavior and gaze shifts during the code review, we employed measurements of *Euclidean distance* and *velocity* derived from consecutive gaze samples. In addition, each raw data point was mapped in real-time to a line in the source code, expressed with the absolute line number. The mapped line numbers were often missing because of low eye-tracking data quality; therefore the features related to code line numbers (e.g. transitions and dwell times) were omitted from the analysis.

Raw TouchMouse data was recorded in form of a 2D grid, representing the surface of the mouse and the capacitance of the reviewers' hands. The 2D information was processed into two components – a sum of the capacitive pixels (TouchMouseSum) and a number of fingers detected from the grid (TouchMouseCount) (Hernandez et al., 2014).

Pre-processed data series from each sensor were sliced with two second time window with no overlap. To characterize the signals' fluctuations within the observed 5 minutes prior to the end of task, a battery of statistical features (i.e. *mean*,*median*, *variance*, *minimum*, *maximum*, *sum*) were computed for each signal in the 2-second data slice (see Table 2). The final feature set contained 55 features (12 from GSR (phasic and tonic component: 2x6), 33 from eye gaze (eye-gaze distance: 3x6; eye-gaze velocity: 3x5), and 10 from TouchMouse (TouchMouse SUM and COUNT: 2x5).

4.3. Machine learning

For overall affect recognition, we investigated how features derived from the physiological signals predicted the reviewer's affect after the code review. Target labels were retrieved from the PAM questionnaires, where each cell in the grid corresponded to the level of valence and arousal (1-4) (Pollak et al., 2011). Figure 3 illustrates the distribution of valence and arousal ratings after the review. The valence and arousal ratings were binarized so that valence was either positive (PAM horizontal score 3 or 4) or negative (PAM horizontal score 1 or 2), and arousal was either low (PAM vertical score 1 or 2) or high (PAM vertical score 3 or 4).

Some affect components can develop fast in time, such as increased arousal as a response to an unexpected surprise. However, other affect components can develop slowly and require time to build up (Ekman and Davidson, 1994; Figner et al., 2011). In this work, we aim to predict the outcomes after the task as the slowly-developed affective states and we explore last five minutes before the end of the task. Code reviews shorter than five minutes were omitted from the analysis. The final dataset consisted of 3900 feature vectors with 55 features.

Recognition performance was evaluated using a Random Forest classifier because of its ability to handle large datasets and its built-in feature selection. Classifier parameters were first optimized using a random grid search with Area Under the ROC Curve (AUC) as the optimization criterion. Next, the classifier was validated with selected parameters in 5x5 crossvalidation. In each fold, the feature set was randomly shuffled and split with stratified sampling to sustain the original class imbalance. Class distribution in the training folds were balanced using the SMOTE approach (Chawla et al., 2002); class distribution in testing folds remained imbalanced[5]. In this work, we report on average accuracy, F1-score, true positive and true negative rates averaged over the testing sets.

5. Results

We report on three primary findings. First, we evaluate participants' affect after the code review task and how external factors potentially contributed to the resulting affective state. Next, we report on affect presented in the written code review comments and perform a regression of affect presence with comment typing characteristics. Last, we discuss recognition performance of valence and arousal-based classification that was trained using physiological features.

5.1. Self-assessed affect after the code review task

After each task, participants assessed their affective state using the Photographic Affect Meter (PAM scale) (Pollak et al., 2011) illustrated in Figure 2. As seen in Figure 3, participants' affect after the task was skewed towards positive valence (on the x-axis), yet showed fairly balanced arousal (on the y-axis).

We hypothesized that numerous effects could impact reviewer's affective state. We expected that covariates such as the long-term affect prior to the experiment (measured as sum of positive and negative affect scores in PANAS), familiarity with the code, seniority of the programmer, and task duration and difficulty (measured as a sum of NASA TLX scores) could impact reviewer's affect. Table 1 summarizes the dependent and independent variables.

We examined the independent variables first using Pearson's correlation. The PANAS component of negative affect was positively correlated with the task duration ($r = 0.365$, $p = 0.037$) and task difficulty ($r = 0.412$, $p = 0.017$). Task duration was also positively correlated with the task difficulty ($r = 0.411$, $p = 0.017$). So, we removed these two covariates to fit the linear regression.

After controlling for covariates (see Table 3), only positive long-term affect prior to the experiment (PANAS) predicted participant's valence after the task (B=0.062, p=0.026). Neither the reviewer's long-term negative affect, familiarity with the code, nor seniority to the code author were able to statistically predict the reviewer's affect after the task.

5.2. The effect of affect on typing speed

Next, we explored whether the emotionality of the comment reflected conventional metrics such as typing duration and typing speed. From the distributions in Figure 4, it was apparent that comment emotionality (white) was associated with the increased mean typing duration and variance; however, this distribution did not take into account the difference explained by the comment

[5] Example scripts are available at https://github.com/hanav/PandasEye

Fig. 2. An example of Photographic Affect Meter (PAM scale).

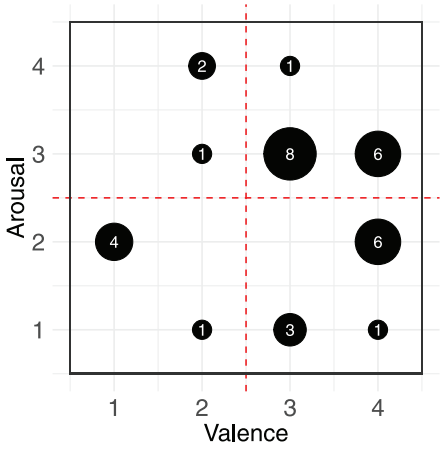

Fig. 3. Distribution of reviewers' valence (horizontal) and arousal (vertical) after their task. The location of the points presents four affective quadrants of the PAM scale. The radius of the points corresponds to the number of participants who reported a particular state.

Table 3
Factors contributing to affect reported after the task. Logistic regressions revealed that only long-term positive affect (assessed by PANAS) was significantly associated with valence after the task.

	Dependent variable			
	Valence		Arousal	
Predictors	*Estimates*	*p*	*Estimates*	*p*
(Intercept)	1.72	0.146	1.76	0.114
PANAS PA	**0.06**	**0.026**	0.02	0.465
PANAS NA	-0.01	0.716	-0.02	0.442
Code familiarity	-0.85	0.132	0.23	0.66
Reviewer's seniority	0.08	0.736	0.12	0.567
Observations	33		33	
R2 / adjusted R2	0.212 / 0.099		0.044 / -0.092	

length. Thus, we evaluated the predictive power of typing-related metrics using a linear model with mixed effects.

The fitted model revealed that the typing duration in an average-length comment was 23.21 seconds. Emotionality increased it significantly by 12.07s (Std.error = 4.00, $F_{1,218}$ =13.49, p-value=0.003**). The average effect of comment length on the duration was 0.24 seconds/character, and emotionality increased it significantly by 0.17 seconds/character (Std.error 0.06, p-value=0.007**) to 0.41 seconds/character. Table 4 reports on the estimated parameters in detail. What stands out is that the standard deviation in typing duration between reviewers was 5.70 seconds, and the variation in the duration per character had standard deviation of 0.03 seconds per character.

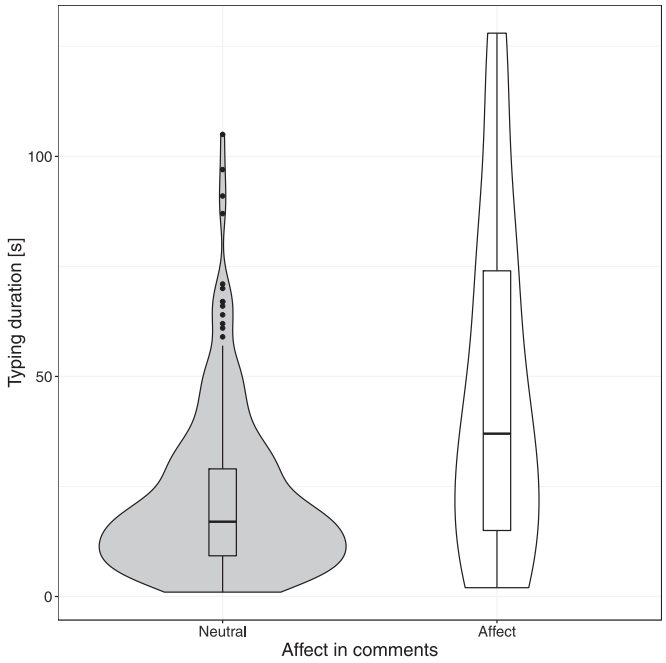

Fig. 4. Typing duration with respect to the comment emotionality. Comments with a neutral tone were produced in shorter time, while comments with emotionality required longer time and presented higher variability.

Table 4
Parameter estimates of the Linear mixed effect model for comment duration in equation (4.1).

Fixed part			
	Estimate	Std. Error	p-value
β_0: Intercept	23.21	1.34	0.000***
β_1: Emotion	12.07	4.00	0.003**
β_2: Comment length	0.24	0.01	0.000***
β_3: Emotion: Comment length	0.17	0.06	0.0069
Random part and residual			
$var(b_i^{(1)})$	5.70^2		
$var(b_i^{(2)})$	0.0306^2		
$cor(b_i^{(1)}, b_i^{(2)})$	0.998		
δ	0.989		
σ^2	0.498^2		

5.3. Predicting affect components from indirect physiology

The predictive power of our multimodal features was evaluated for both affect components separately (valence and arousal) using a Random Forest classifier and a 5x5 kFold shuffled cross validation. Table 5 summarizes the performance of the model achieved on cross validation test sets for all features (modality fusion) and features extracted from individual sensors. Baseline performance was obtained on the full feature set using a default dummy classifier.

In recognition of valence and arousal after the task, the best performance was achieved using the fusion of the modalities. The overall model of valence performed higher (*accuracy* = 90.0%, *F1score*=0.937) compared to the model of arousal (*accuracy* = 83.9%, *F1score* = 0.856); the model of valence predicted better the positive valence labels (*TPR* = 0.957) than the negative valence (*TNR* = 0.695), suggesting that the directionality of valence is somewhat reflected in the signals. The model of arousal predicted the high arousal (*TPR* = 0.853) higher than the negative arousal (*TNR* = 0.823); however, the differences were minor, suggesting that the polarity of arousal was easily distinguishable from the employed physiological signals.

Considering the individual modalities, eye gaze alone performed better than other modalities both in recognition of valence (*accuracy* = 85.8%, *F1score* = 0.912) and arousal (*accuracy* = 76.6%, *F1score* = 0.785). While scoring higher in favor of valence, however, the gaze-based classifier delivered a balanced performance in favor of arousal. Of all modality combinations, touch pressure predicted negative valence better (*TNR* = 0.734) compared to positive valence (*TPR* = 0.605).

5.4. Predicting affect in time

It is a reasonable assumption that affect builds up during the task, given the fact that important events occur during interaction with the code. Therefore, we hypothesized that recognition of the target labels should be harder earlier in the dataset. In other words, should such a system be implemented in real-life, it is important to understand whether early recognition based on historical data would perform as well as recognition based on more recent inputs. To test this hypothesis, we extracted the same feature sets from the beginning of the task, set the same labels as measured after the task, and repeated the analyses.

As illustrated in Table 6, recognition results on past data were approximately 4% worse in both fusion models compared to the models based on recent data from the end of the task. The largest differences were observed in the galvanic skin response both in valence($\Delta accuracy$ = 5.12%, $\Delta F1score$ = 0.043) and arousal ($\Delta accuracy$ = 7.27%, $\Delta F1score$ = 0.051).

6. Discussion

Affect in collaborative tasks in general, and in code review in particular, have important consequences for team performance. Indeed, happy programmers have been observed to be more productive (Graziotin et al., 2018). Negative affect, on the other hand, may create an obstacle towards professional conduct, or performance of a task (Gachechiladze et al., 2017). With the prevalence

Table 5
Classification of valence and arousal at the end of the task using last five minutes of data. The best performance was achieved when including all features (modality fusion). Baseline results were obtained using the dummy classifier from Scikit-learn in training.

	Measure	Baseline	GSR	Eye gaze	Touch pressure	Modality fusion
Valence	ACC	0.490	0.686	0.858	0.634	0.900
	F1	0.599	0.789	0.912	0.719	0.937
	AUC	0.493	0.670	0.870	0.761	0.937
	TPR	0.488	0.754	0.946	0.605	0.957
	TNR	0.499	0.446	0.547	0.734	0.695
Arousal	ACC	0.517	0.629	0.766	0.697	0.839
	F1	0.540	0.664	0.785	0.753	0.856
	AUC	0.518	0.682	0.860	0.771	0.922
	TPR	0.507	0.654	0.763	0.826	0.853
	TNR	0.529	0.597	0.771	0.532	0.823

Table 6
Classification of valence and arousal at the beginning of the task using first five minutes of data.

	Measure	Baseline	GSR	Eye gaze	Touch pressure	Modality fusion
Valence	ACC	0.488	0.635	0.822	0.678	0.855
	F1	0.597	0.746	0.890	0.770	0.910
	AUC	0.490	0.596	0.826	0.747	0.870
	TPR	0.486	0.688	0.927	0.692	0.950
	TNR	0.493	0.448	0.450	0.632	0.518
Arousal	ACC	0.515	0.556	0.723	0.684	0.798
	F1	0.539	0.612	0.750	0.701	0.822
	AUC	0.517	0.562	0.806	0.773	0.884
	TPR	0.505	0.626	0.740	0.665	0.831
	TNR	0.528	0.467	0.702	0.707	0.756

of computer-mediated communication during code review, textual comments do not effectively transmit the subtle, yet significant, social and behavioral cues necessary for correct affect recognition.

This is an interesting and important finding from the perspective of prior research that links the developer's happiness and satisfaction with their work to factors such as higher performance, engagement, and problem solving (Müller and Fritz, 2015; Graziotin et al., 2015; 2018). Our findings suggest that also long-term affect contributes to this equation. While more empirical support needs to be collected in the future, it has become clear that theories of affect in software engineering and models describing well-being at software development work need to expand beyond simplistic measures of task-related affect.

Further research is thus needed to investigate the relationships between long-term affect and task-related affect, and examine other long-term factors, such as developer's well-being, health status, and personal life situations along with work satisfaction and performance. In addition, it would be interesting theorectically to compare whether long-term or short-term factors have more impact on post-task affect.

These and future findings inform professional software development environments about the plausible factors which impact developers' performance that are worth enhancing and those that are beyond a company's reach. Finally, these findings need to be taken with a grain of salt. Although they would be valuable and informative for current theories on (un)happiness in software development, repetitive qualitative assessments of long-term and short-term affect would likely be infeasible to incorporate into daily work practice. This is why we explored the feasibility of nonverbal signals for affect detection.

The best performance in affect recognition from nonverbal signals currently comes from facial and prosodic expressions (D'mello and Kory, 2015). Although these signals are excellent candidates for affect detection (Zeng et al., 2008), recording and analzying people's daily affect raises privacy issues and is potentially challenging for long-term recording in a corporate environment. Consequently, it is the conventional behavioral signals and indirect physiological expressions, such as galvanic skin response, eye gaze, or touch pressure, that present more suitable candidates for affect recognition. They do not require effort and collaboration, cannot be easily controlled by the users, and cannot be directly interpreted by an external observer, thus do not violate the sense of one's privacy.

In this work, we explored opportunities of physiological signals and their relation to one's affect, and measured these signals *in situ* with minimal interference to professional source code review practice. In our follow-up analysis using a machine-learning framework, we assessed feasibility of these signals for automatic affect detection.

In the first question, we examined how the aspects specific to the task (i.e. reviewer's seniority and familiarity with the code) and long-term affect impact the affect after the code-review task. Only positive long-term affect was associated with after-task valence, suggesting that participants' well-being prior to the code review contributes to their level of happiness after the task.

This is an interesting finding from the perspective of prior research that linked developer's happiness with their work to factors such as higher performance, engagement, and problem solving Graziotin et al. (2018); Müller and Fritz (2015). Our findings suggest that the long-term affect also contributes to this equation. Given our small sample size, however, we would be cautious to claim that the long-term happiness dictates the reviewer's after-task happiness. Further research is needed to investigate the role of the long-term affect concerning the task-related affect. We need to examine long-term factors, such as developer's well-being, health state, or personal life, along with work satisfaction and performance. In addition, it would be interesting to compare whether long-term or short-term factors have a greater impact on post-task affect. Those findings would help people in professional environments identify which factors could plausibly enhance developers' performance and which factors are beyond their reach.

The second question in this paper was the influence of the presence of emotionality in the comment on a reviewer's commenting behavior, such as duration of typing the comment and the comment length in terms of characters. To answer this question, we first evaluated whether reviewers' comments contained any recognizable affect at all. When two independent raters manually annotated the comment base, comments containing emotion represented a minority of the comment base (bellow 10%). This was expected and is in line with prior research. This finding thus further supports the need for other means of affect recognition in code review.

As noted by Lutchyn et al. Lutchyn et al. (2015), corporate culture inhibits certain emotions as inappropriate or undesirable in the workplace, and workers are expected to regulate their affective expressions. While the overt expressions of affect can be voluntarily suppressed, involuntary behaviors, such as typing, do communicate affect, as we show in this work.

Our results suggest that comments with affect required significantly more time to type, independent of comment length or task order. Specifically, comments with emotional content increased the average typing duration by 12.70 seconds in total, or 0.17 seconds per character. These findings suggests that using models of typing behavior for automatic affect detection are viable. However, variability in this metric was high, making the commenting behavior metrics infeasible for effective affect recognition.

Our third question considered the extent to which physiological signals correspond to genuine affective states in *in situ* code review tasks. In recognition of valence and arousal, a fusion of three modalities delivered the best performance, much higher than the baseline, in favor of recognition of a participant's valence. Though recognition between high and low arousal was fairly balanced, recognition performance of positive and negative valence was skewed towards the positive valence.

When comparing performance of individual modalities, eye gaze signals delivered the greatest overall recognition performance. Touch pressure delivered equal recognition for high and low arousal in the recognition of arousal, which corresponds with findings of Hernandez et al. Hernandez et al. (2014). Models utilizing galvanic skin response scored the lowest out of the three sensors.

Overall, certain aspects of affect are more difficult to detect than others and not all approaches and sensors are equally suited for affect detection *in situ*. Of the long-term and task-related aspects, only long-term positive affect is predictive of post-task valence, though not arousal. The presence of emotionality in comments is associated with the time required to type a comment, but not with the length of the comment. And finally, fusing the physiological signals performs best overall for post-task affect, outperforming individual sensors.

6.1. Implications and Future work

The leading challenge in remote, computer-mediated, asynchronous communication arises when affective information is undetected or misinterpreted by the other party (Ebert et al., 2019). The results presented here provide concrete implications for both research and industry, and lay the foundation for investigations in real-life professional software development.

One concrete recommendation based on this work is to employ a fusion of eye-gaze, touch-sensing, and GSR sensors. The detailed evaluation of the feasibility of these three modalities for affect recognition introduces grounds for their joint application in the industrial settings.

Based on these sensors, we envision a novel form of implicit affect-sensing system that continuously monitors affect during code review. The low-cost sensing setups, our results show, can be successfully embedded into the development and code-review environments without many modifications. The methods and the sensors introduced here present a necessary framework to further the understanding of the link between emotions and work in software development teams.

As Girard et al. Girardi et al. (2018) proposed in their benchmark study in software development, understanding other's affective state is beneficial at multiple social and organizational levels. In daily work, intelligent multimodal affect recognition could allow reviewers and developers to better communicate the meaning of the comment and assist in setting of the importance of written messages.

In this study we modeled the affective states of the reviewer, and by doing so we set the stage for future work to identify how developers emotionally experience their reviewers' comments. Future research will focus on questions around how to meaningfully communicate these recognized affective states (Picard, 1997; Barral et al., 2016), and also on how affect-enhanced code review improves the communication between remote software development teams.

Future modeling approaches can also extend our findings to other factors occurring in professional software development, such as confusion and misinterpretation, to the role of culture (Elfenbein and Ambady, 2002), and their relationship to the productivity of remote teams.

6.2. Threats to validity

As in any *in situ* study, our work is not exempt from limitations. In this work, we purposefully employed affordable sensors and situated the study in daily code review activities to ensure high ecological validity, which inherently introduced several limitations.

The results we report here on affect in comments were limited by the sample size and class imbalances. A larger set of annotated comments would be required to validate the results. Obtaining a manually annotated database of code comments, preferably project and language specific, however, is one of the current challenges in the sentiment analysis research (Islam and Zibran, 2018; Basile et al., 2018). Although beneficial, creating such a database would require considerable resources of multiple project-knowledgeable raters.

This study trades off data quality, affordability of sensors, and optimal data collection conditions. With respect to use of eye trackers, we visited each participant on-site, and could not ensure even illumination of the offices, nor enforce the participants to sit an optimal distance from the eye tracker, as recommended for eye-tracking experiments (Holmqvist et al., 2011).

Similarly, readings of GSR are influenced by factors such as environmental temperature, physical activity, and individual differences in physiology (Braithwaite et al., 2013). We did not calibrate the temperature in the office, nor enforce recommended physical exercise prior to the experiment to increase the accuracy of the GSR sensor (Braithwaite et al., 2013). In addition, in our study we observed the code review task eliciting predominantly medium levels of arousal. Taken together, we conclude that in this case, GSR was less sensitive to subtle changes in arousal and, therefore, less suitable for arousal recognition.

Due to the unrestricted settings and nature of the sensors, however, we expected challenges with data collection and evaluations, and compensated for them in form of careful and robust data processing, filtering, and selection.

Finally, one potential threat to validity is related to possible inter-correlation of the multimodal features. In such a case, the predictive power of the model could be reduced. Further research could examine feature importance also through additional methods such as exhaustive recursive search to reveal such dependencies. However, such an approach would demand additional computational power and an adequately small feature set.

7. Conclusion

In code review, a reviewer argues about the validity of the code. Why was a particular piece of code written in the particular way? How does it fit into its particular position in the current project hierarchy? Is it suited to the project best practices? Does it violate software efficiency? These are just a few. The reviewer's internal cognitive and emotional states related to their code reviews, however, are unknown to the author of the code and rarely propagate to the reviewer's feedback, as we observed in the current study.

In professional software development settings, code review is a beneficial practice to improve code quality, share best practices among colleagues, and reduce the resources needed in product testing. However, when the code is reviewed using computer-mediated communication tools, the reviews lack important socio-cognitive cues that are crucial for efficient team functioning.

In this work, we investigated the potential of unobtrusive affect sensing using biometric sensors for purposes of enhancing code review. We grounded our investigation using Linear Mixed Effect models and machine learning to capture affect during source code reviews in a real-life, *in situ* data collection. With minimal interference to professional source code review practice, we collected physiological signals related to affective states and performed modeling and analysis to automatically detect the reviewers' affect.

Authentic affect in the written reviews was significantly associated with increased typing duration of the comment. Genuine affect after the task was recognizable from employed biometric sensors that were installed on site.

Intelligent multimodal affect recognition in code review opens up new research directions and applications. The next generation of code review tools could utilize affect recognition to better communicate detected affect in code review. Future research on computer-mediated team collaboration activities could extend our study to investigate affective information received by author developers, explore the discrepancy between the reviewer's genuine affect and developer's perceived affect from the written reviews, and identify the extent to which intelligent affect-awareness embedded in the code review might remedy understanding and communication challenges.

Acknowledgment

The work was supported by a Microsoft internship and by the Academy of Finland grant No. 305199.

Appendix

Table 7
Confusion matrix of first-round raters' coding of comments emotionality.

	R2 (positive)	R2 (neutral)
R1 (positive)	17	82
R1 (neutral)	4	156

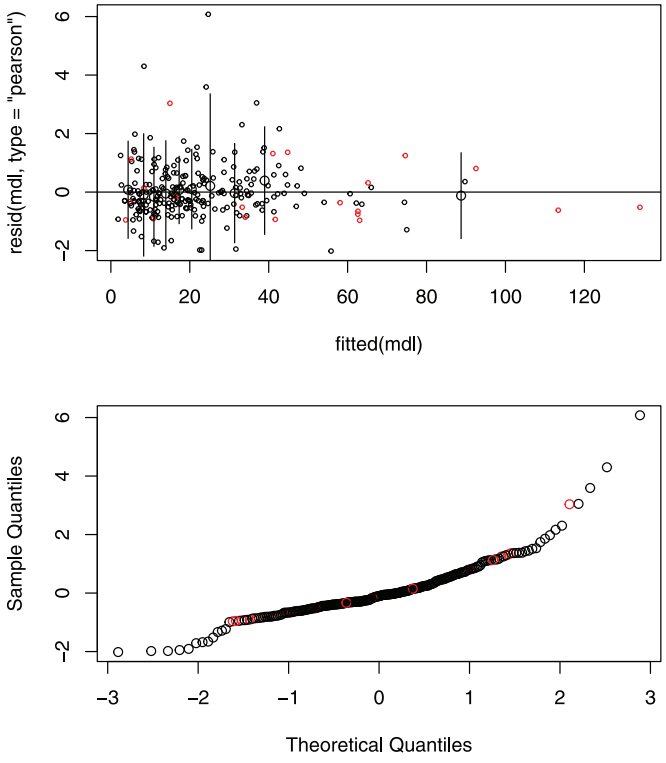

Fig. 5. Diagnostic plots of the final model. The residual and Q-Q plot are based on Pearson residuals. Neutral comments are in depicted in black, comments with emotionality are illustrated in red.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jss.2019.110434.

References

Ahmed, T., Bosu, A., Iqbal, A., Rahimi, S., 2017. Senticr: a customized sentiment analysis tool for code review interactions. In: Proceedings of the 32nd ieee/acm international conference on automated software engineering. IEEE Press, pp. 106–111.

Bacchelli, A., Bird, C., 2013. Expectations, outcomes, and challenges of modern code review. In: Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, Piscataway, NJ, USA, pp. 712–721.

Barral, O., Kosunen, I., Ruotsalo, T., Spapé, M.M., Eugster, M.J., Ravaja, N., Kaski, S., Jacucci, G., 2016. Extracting relevance and affect information from physiological text annotation. User Modeling and User-Adapted Interaction 26 (5), 493–520.

Basile, V., Novielli, N., Croce, D., Barbieri, F., Nissim, M., Patti, V., 2018. Sentiment polarity classification at evalita: Lessons learned and open challenges. IEEE Transactions on Affective Computing doi:10.1109/TAFFC.2018.2884015. 1–1

Bates, D., Mächler, M., Bolker, B., Walker, S., 2015. Fitting linear mixed-effects models using lme4. Journal of Statistical Software 67 (1), 1–48. doi:10.18637/jss.v067.i01.

Braithwaite, J.J., Watson, D.G., Jones, R., Rowe, M., 2013. A guide for analysing electrodermal activity (eda) & skin conductance responses (scrs) for psychological experiments. Psychophysiology 49, 1017–1034.

Burns, A., Doheny, E.P., Greene, B.R., Foran, T., Leahy, D., O'Donovan, K., McGrath, M.J., 2010. Shimmer: an extensible platform for physiological signal capture. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE, pp. 3759–3762.

Calefato, F., Lanubile, F., Maiorano, F., Novielli, N., 2018. Sentiment polarity detection for software development. Empirical Software Engineering 23 (3), 1352–1382.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Coyne, J., Sibley, C., 2016. Investigating the use of two low cost eye tracking systems for detecting pupillary response to changes in mental workload. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 60. Sage Publications Sage CA: Los Angeles, CA, pp. 37–41.

De Choudhury, M., Counts, S., 2013. Understanding affect in the workplace via social media. In: Proceedings of the 2013 conference on Computer supported cooperative work. ACM, pp. 303–316.

Dewan, P., 2015. Towards emotion-based collaborative software engineering. In: Cooperative and Human Aspects of Software Engineering (CHASE), 2015 IEEE/ACM 8th International Workshop on. IEEE, pp. 109–112.

D'mello, S.K., Kory, J., 2015. A review and meta-analysis of multimodal affect detection systems. ACM Computing Surveys (CSUR) 47 (3), 43.

Ebert, F., Castor, F., Novielli, N., Serebrenik, A., 2018. Communicative intention in code review questions. In: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, pp. 519–523.

Ebert, F., Castor, F., Novielli, N., Serebrenik, A., 2019. Confusion in code reviews: Reasons, impacts, and coping strategies. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, pp. 49–60.

Ekman, P.E., Davidson, R.J., 1994. The nature of emotion: Fundamental questions.. Oxford University Press.

Elfenbein, H.A., Ambady, N., 2002. On the universality and cultural specificity of emotion recognition: a meta-analysis.. Psychological bulletin 128 (2), 203.

Ericsson, K.A., Simon, H.A., 1993. Protocol analysis. MIT press Cambridge, MA.

Fagan, M.E., 1999. Design and code inspections to reduce errors in program development. IBM Systems Journal 38 (2/3), 258.

Figner, B., Murphy, R.O., et al., 2011. Using skin conductance in judgment and decision making research. A handbook of process tracing methods for decision research 163–184.

Fritz, T., Begel, A., Müller, S.C., Yigit-Elliott, S., Züger, M., 2014. Using Psycho-physiological Measures to Assess Task Difficulty in Software Development. Proceedings of the 36th International Conference on Software Engineering 402–413.

Gachechiladze, D., Lanubile, F., Novielli, N., Serebrenik, A., 2017. Anger and its direction in collaborative software development. In: Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER), 2017 IEEE/ACM 39th International Conference on. IEEE, pp. 11–14.

Gibaldi, A., Vanegas, M., Bex, P.J., Maiello, G., 2017. Evaluation of the tobii eyex eye tracking controller and matlab toolkit for research. Behavior research methods 49 (3), 923–946.

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.

Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P., 2018. What happens when software developers are (un) happy. Journal of Systems and Software 140, 32–47.

Graziotin, D., Wang, X., Abrahamsson, P., 2013. Are happy developers more productive? In: International Conference on Product Focused Software Process Improvement. Springer, pp. 50–64.

Graziotin, D., Wang, X., Abrahamsson, P., 2015. Do feelings matter? on the correlation of affects and the self-assessed productivity in software engineering. Journal of Software: Evolution and Process 27 (7), 467–487.
Hancock, J.T., Landrigan, C., Silver, C., 2007. Expressing emotion in text-based communication. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, pp. 929–932.
Hart, S.G., Staveland, L.E., 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: Advances in psychology, 52. Elsevier, pp. 139–183.
Herbsleb, J.D., Klein, H., Olson, G.M., Brunner, H., Olson, J.S., Harding, J., 1995. Object-oriented analysis and design in software project teams. Human–Computer Interaction 10 (2-3), 249–292.
Herbsleb, J.D., Mockus, A., 2003. An empirical study of speed and communication in globally distributed software development. IEEE Transactions on software engineering 29 (6), 481–494.
Hernandez, J., Paredes, P., Roseway, A., Czerwinski, M., 2014. Under pressure: sensing stress of computer users. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, pp. 51–60.
Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., Van de Weijer, J., 2011. Eye tracking: A comprehensive guide to methods and measures. Oxford University Press, London.
Huang, M.X., Kwok, T.C., Ngai, G., Chan, S.C., Leong, H.V., 2016. Building a personalized, auto-calibrating eye tracker from user interactions. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 5169–5179. doi:10.1145/2858036.2858404.
Islam, M.R., Zibran, M.F., 2018. Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text. Journal of Systems and Software 145, 125–146.
Kuutila, M., Mäntylä, M., Claes, M., Elovainio, M., Adams, B., 2018. Using experience sampling to link software repositories with emotions and work well-being. arXiv preprint arXiv:1808.05409.
Lutchyn, Y., Johns, P., Roseway, A., Czerwinski, M., 2015. Moodtracker: Monitoring collective emotions in the workplace. In: Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on. IEEE, pp. 295–301.
Mäntylä, M.V., Graziotin, D., Kuutila, M., 2018. The evolution of sentiment analysis a review of research topics, venues, and top cited papers. Computer Science Review 27, 16–32.
McKinney, W., 2010. Data structures for statistical computing in python. In: van der Walt, S., Millman, J. (Eds.), Proceedings of the 9th Python in Science Conference, pp. 51–56.
Müller, S.C., Fritz, T., 2015. Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. Proceedings - International Conference on Software Engineering 1, 688–699.
Novielli, N., Girardi, D., Lanubile, F., 2018. A benchmark study on sentiment analysis for software engineering research. In: 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR). IEEE, pp. 364–375.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.
Peng, F., Li, C., Song, X., Hu, W., Feng, G., 2016. An eye tracking research on debugging strategies towards different types of bugs. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), 2, pp. 130–134. doi:10.1109/COMPSAC.2016.57.
Picard, R.W., 1997. Affective Computing. MIT Press, Cambridge, MA, USA.
Picard, R.W., 1999. Affective computing for hci. Procs. 8th HCI International on Human-Computer Interaction: Ergonomics and User Interfaces 829–833.
Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., R Core Team, 2017. nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-131.
Pollak, J., Adams, P., Gay, G., 2011. Pam: A photographic affect meter for frequent, in situ measurement of affect. In: Proceedings of CHI, pp. 725–734.
R Core Team, 2015. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria.
Riordan, M.A., Trichtinger, L.A., 2017. Overconfidence at the keyboard: Confidence and accuracy in interpreting affect in e-mail exchanges. Human Communication Research 43 (1), 1–24.
Sano, A., Johns, P., Czerwinski, M., 2017. Designing opportune stress intervention delivery timing using multi-modal data. In: Affective Computing and Intelligent Interaction (ACII), 2017 Seventh International Conference on. IEEE, pp. 346–353.
Sarker, F., Vasilescu, B., Blincoe, K., Filkov, V., 2019. Socio-technical work-rate increase associates with changes in work patterns in online projects. In: Proceedings of the 41st International Conference on Software Engineering. IEEE Press, pp. 936–947.
Schmidt, P., Reiss, A., Dürichen, R., Van Laerhoven, K., 2018. Labelling affective states in the wild: Practical guidelines and lessons learned. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers. ACM, pp. 654–659.
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. Journal of Systems and Software 143, 59–70.
Schulze, J., Krumm, S., 2017. The 'virtual team player' a review and initial model of knowledge, skills, abilities, and other characteristics for virtual collaboration. Organizational Psychology Review 7 (1), 66–95.
Shu, L., Xie, J., Yang, M., Li, Z., Li, Z., Liao, D., Xu, X., Yang, X., 2018. A review of emotion recognition using physiological signals. Sensors 18 (7), 2074.
Storey, M.-A., Zagalsky, A., Singer, L., German, D., et al., 2017. How social and communication channels shape and challenge a participatory culture in software development. IEEE Transactions on Software Engineering (1) 185–204.
Tang, H., Tan, S., Cheng, X., 2009. A survey on sentiment detection of reviews. Expert Systems with Applications 36 (7), 10760–10773.
Uwano, H., Nakamura, M., Monden, A., Matsumoto, K.-i., 2006. Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement. Eye tracking research & applications (ETRA) 133–140.
van der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The numpy array: A structure for efficient numerical computation. Computing in Science Engineering 13 (2), 22–30. doi:10.1109/MCSE.2011.37.
Watson, D., Clark, L., Tellegan, A., 1988. Development and validation of brief measures of positive and negative affect. Journal of Personality and Social Psychology 54 (6), 1063–1070.
Wrobel, M.R., 2013. Emotions in the software development process. In: 2013 6th International Conference on Human System Interactions (HSI). IEEE, pp. 518–523.
Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S., 2008. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE transactions on pattern analysis and machine intelligence 31 (1), 39–58.

Hana Vrzakova is a Postdoctoral Researcher at the Institute of Cognitive Science, University of Colorado Boulder, CO, USA. In her research, she bridges fields of eye tracking, HCI, and data science, in order to computationally understand visual attention in human-computer and human-human interaction. Her research spans the contexts where understanding individual's and group's attention can inform the design of intelligent, pervasive, and proactive interfaces.

Andrew Begel is a Principal Researcher in the Ability group at Microsoft Research in Redmond, WA, USA. He received his Ph.D. in Computer Science from the University of California, Berkeley in 2005. Andrew focuses on helping people on the autism spectrum achieve employment and facilitate social interaction. Andrew also explores evolving job roles in the software industry and studies the growing impact of AI technologies on software engineering.

Lauri Mehtätalo is Professor in Applied Statistics at University of Eastern Finland. He received his Ph.D. in Forest Sciences at the University of Joensuu in 2004. He is involved in numerous research projects as statistical consultant, especially in the use of mixed-effects models in various fields and research problems.

Roman Bednarik is an Associate Professor in the Interactive Technologies group at the University of Eastern Finland. Roman is interested in understanding and modeling of human behavior, cognition, learning, and interaction with computerized systems. He develops and applies eye-tracking technologies in a variety of interactive domains.