Fagerholm, Fabian; Fritz, Thomas

Biometric Measurement in Software Engineering

Published in:
Contemporary Empirical Methods in Software Engineering

DOI:
10.1007/978-3-030-32489-6_6

Published: 28/08/2020

Document Version
Peer reviewed version

Please cite the original version:
Fagerholm, F., & Fritz, T. (2020). Biometric Measurement in Software Engineering. In *Contemporary Empirical Methods in Software Engineering* (pp. 151-172). Springer. https://doi.org/10.1007/978-3-030-32489-6_6

Biometric Measurement in Software Engineering

Fabian Fagerholm and Thomas Fritz

Abstract Biometric sensor technology provides new opportunities to measure physiological changes in the human body that can be linked to various psychological processes. In software engineering, these biometric measurements can be used to gain insights on fundamental cognitive and emotional processes of software developers while they are working. In addition, biometric measures may be used to provide better and more instantaneous tool support for developers, for instance by preventing defects from being introduced in the code or supporting focused work. In this book chapter, we motivate the use of biometric measurements, introduce common types of biometric sensors and measures, discuss how to choose the right set of them and considerations for analyzing the collected data. We also discuss work in the area of software engineering and recommend further reading.

1 Introduction

Software is built by humans. Software developers are the ones who develop and evolve code, that elicit requirements, test the software, and talk to their teammates to coordinate. Yet, traditionally, research has focused to a large extent on normative processes and artefacts – how developers ought to develop software, the digital objects developers have created or modified, measuring their output, and collecting data from software repositories. While this focus on ideal work processes and developers' output can provide interesting and relevant insights, it falls short when the goal is to better understand the humans in the process, such as the cognitive demands and emotions they experience, and the individual differences between developers while

Fabian Fagerholm
Aalto University, Finland & Blekinge Institute of Technology, Sweden,
e-mail: fabian.fagerholm@aalto.fi

Thomas Fritz
Department of Informatics, University of Zurich, Switzerland, e-mail: fritz@ifi.uzh.ch

they create and evolve the output data. Especially since these human aspects can have a significant effect on the output and its quality, such as a higher cognitive load leading to a higher error rate (Sweller, 1988; Ayres, 2001), the better we understand the human in the process, the better we can support the software development endeavour, and the better software quality we can achieve.

There are many ways we can measure the cognitive and emotional processes that are active while humans develop software. Some of these provide indirect means to gain more understanding, such as self-report measurements of happiness (Graziotin et al., 2018) or objective performance on cognitively demanding tasks. Recent advances in biometric (*aka.* psycho-physiological) sensor technology offer new opportunities to collect and examine a wide variety of direct, detailed data on a human and her cognitive and emotional states while she is working and developing software. The underlying idea is that a human's psychological states are linked to her physiological processes and that these physiological processes can be measured using biometric sensors. Research in psychology and other fields has already investigated and correlated certain biometric measures, including skin-, heart-, eye- and brain-related metrics, with a human's cognitive and emotional states. For instance, researchers have found that pupil size and electro-dermal activity (EDA) can be linked to cognitive load (Wilson, 1992; Richter et al., 1998; Setz et al., 2010; Haapalainen et al., 2010; Iqbal et al., 2004).

Research in software engineering is also starting to take advantage of biometric measurements using a variety of sensor technology to examine cognitive processes among software developers. These studies range from the use of an eye-tracker to capture a developer's eye fixations when reading or navigating code (Crosby and Stelovsky, 1990; Bednarik and Tukiainen, 2006; Sharif et al., 2012; Kevic et al., 2015), EDA wrist bands, EEG sensors, or chest straps to captureskin conductivity as well as brain- and heart-related metrics to assess mental load and the experienced difficulty of the code (Nakagawa et al., 2014; Fritz et al., 2014; Müller and Fritz, 2016), all the way to the use of functional magnetic resonance imaging (fMRI) and near infrared spectroscopy (NIRS) to examine brain activation patterns during program comprehension (Siegmund et al., 2014; Ikutani and Uwano, 2014).

As biometric sensor technology becomes less invasive, easier to integrate into the developer's work, cheaper, and more accurate, we are now able to capture more fine-grained biometric data of software developers in real-time and in real-life environments. These advances will not only allow us to better understand a developer while working, but also to develop and provide more instantaneous support to developers. For instance, by monitoring when a developer is experiencing a high cognitive load, we might be able to reduce interruptions by other coworkers at inopportune moments, or be able to intervene before a developer introduces a defect. Moreover, we will be able to gain a more fundamental understanding of software development that could lead to advances that we cannot imagine today.

Overall, the research results on the use of biometric measurements in software engineering already demonstrate the potential of this data. At the same time, there are still a lot of challenges to address before the use of biometric data can become widely accepted. These challenges range from the exact interpretation of the data, its noisiness, sensor limitations and invasiveness, to privacy concerns of the developers. For instance, low heart rate variability can generally be linked to a person's stress and high cognitive load, yet, when a low heart rate variability is detected, it is not straightforward to differentiate whether this is due to stress, a high cognitive load, or both. Another example is the limitation and noisiness of current low-invasive heart rate sensors that use optical sensing. Since an optical sensor is affected by movements or changes in the environmental lighting, it is difficult to accurately and reliably capture heart rate variability data with such sensors, especially when they are integrated into wristbands and users move around a lot.

This chapter introduces some common types of biometric sensors and measures that have been used in software engineering research, discusses some of the challenges involved in such research, and gives examples of research in the area. Throughout the chapter, we use a motivating example to illustrate how the information in the chapter could be applied in real-life software development.

2 Motivating example

To illustrate the potential value of biometric data, we will sketch out and highlight a few scenarios of a team developing software. This team, which is part of a larger software development organisation, has three developers: *Mary*, a senior developer with 10 years of professional experience in Java backend development; *Joe*, an experienced developer with 5 years of professional experience on full stack development including JavaScript and Java amongst others; and *Sam*, a new hire fresh out of school with some experience in Java and JavaScript development. All three developers are working together to finish the next version of their software application within a month. While the first two weeks were pretty much free of any time pressure, now that they are a week away from the milestone, the time pressure is slowly picking up a bit. This may influence some important psychological processes:

Stress. Joe's eighth-month old son has had a fever for the past week and hasn't been sleeping well. Since Joe and his wife alternate nights to take care of their son, Joe has been getting a lot less sleep at home than usual and therefore is feeling more and more stressed about the approaching milestone at work and getting everything done. As studies in psychology have shown, increased stress often leads to an increased error rate, and the lack of sleep additionally makes it harder for Joe to concentrate and work for extended periods of time.

Stress generally evokes a physiological response, such as higher blood pressure, an increase in perspiration (also known as sweating) or a decrease in the variability of the heart rate. Biometric sensor technology can be used to capture these physiological responses, such as a wristband to measure electrodermal activity, or a chest strap to measure heart rate variability. By using such sensors, one might be able to continuously monitor the stress level of a person and then use this information to support them in their work. Some possible ways of supporting could be to suggest taking more breaks, reassigning tasks, or recommending additional code reviews to lower the chances of errors being committed to the project.

Cognitive Load. Due the time pressure in the current iteration cycle, each developer is supposed to help with any open tasks. Therefore, it happened that Sam picked up a task that requires changes to the Java backend. While Sam is working on the task, he experiences a high cognitive load due to the lack of expertise in Java development and the backend as well as the complexity of the existing code that he has to change.

As studies in psychology have shown, a high cognitive load is often linked to certain physiological responses, such as a higher pupil dilation, less eye blinks and a lower heart rate variability (HRV). By using an eye tracking sensor or a HRV sensor, we might be able to track the cognitive load and determine when the developer is experiencing difficulties and intervene before he introduces a defect. At the same time, by measuring the difficulty that all developers on the team have when they are working on specific parts of the code, we might be able to determine which parts of the code base are more or less challenging, hinting at where the technical debt is higher and where refactoring might provide the biggest benefits.

Availability for interruptions. As in all office environments, the developers of this team are often interrupted by their co-workers from both their own team and teams from other parts of the organisation. When these interruptions happen at inopportune moments, such as a developer being very focused on the task and memorising a lot of relevant context for the task in her head, or the developer being very engaged in the task, the interruptions can lead to a steep increase in the time needed to complete the tasks, heightened frustration for the developer, and an increased error rate, as studies have shown (Bailey et al., 2001; Czerwinski et al., 2000; Mark et al., 2008).

Similarly to cognitive load and stress, this focus on and engagement in the task might express itself in physiological responses that can then be measured using biometric sensor technology. By continuously monitoring the relevant biometric measure, we might be able to indicate to co-workers the developer's availability for interruptions (*aka.* interruptibility) and thereby help to shift the interruptions to more opportune moments.

Common to all of these scenarios is that there are individual differences between the developers that influence the process of developing software and the end result of it. These range from individual characteristics – such as temperament and personality, how the individual is affected by time pressure, and their level of expertise – to social characteristics – such as the overall stress level at work and at home, the amount of support received by co-workers and the organisation at large, and the consequences of success and failure. By only focusing on the output, it is difficult to detect these individual differences that influence both the developers' experience and the outcomes of the development endeavour. By using biometric measurements that can be captured in real-time, we might be able to intervene earlier and better support the human in the process.

3 Biometrics or Psycho-Physiology

Biometric data generally denotes measurements made on some part of the human body. These measurements could, for example, be used to identify and authenticate an individual, such as a fingerprint, a voice, or a DNA sample. In this chapter, we focus more specifically on measurements made on some part of the human body that are linked to various psychological processes, such as performing cognitive tasks or experiencing emotions. This type of biometric data is also often referred to as psycho-physiological data. In the following, we will use the term biometric data and psycho-physiological data interchangeable to denote this kind of data.

Biometric data is usually structured as time-series data, and is analysed for changes in response to a stimuli, such as a task, a piece of code, a diagram, an emotion inducing picture, or something else that has relevance for what is being studied.

3.1 Examples of biometric sensors and measures

Many different physiological measurements are used in research today due to their ability to reveal more about what goes on in a person's mind. In software engineering research, such measures have commonly been used to determine what a person is paying attention to in a cognitively demanding task, how much the task occupies their mental capacity, or to what extent the task triggers an emotional response. Physiological measurements can roughly be categorized by the origin of the measurement: eye-, skin-, brain-, heart-, and breathing-related measurements. Table 1 presents an overview of some of these measures and the psychological states and processes they have been linked to in previous research, predominantly in psychology. In the following we briefly discuss how a few selected physiological measurements and sensors work in principle.

3.1.1 Eye-related Measurements

The inner workings of the mind can be probed by measuring how various eye muscles contract. By taking advantage of this psycho-physiological fact, eye tracking hardware and software have developed to a level where accurate and fine-grained measurements of the eye can be made in a minimally invasive way.

Eye tracking can be divided into three common measurements: eye gaze, pupillary responses, and eyeblink rate. Each of these provide information on what the human mind is doing, and when combined with specific tasks in carefully designed study set-ups, they can be used to gain understanding of where the person is focusing their attention and how their mind is processing the information that the eye sees.

Eye gaze contains information on what a person is looking at. Many studies corroborate the fact that a person's eyes are generally directed towards the object she

Table 1 Overview of several physiological measures and previously linked states and processes (see also (Fritz and Müller, 2016)).

Measure	Previously linked to
Eye-related	
Eye gaze	cognitive load (Ikehara and Crosby, 2005); valence (Carniglia et al., 2012)
Pupillary response	cognitive & mental load (Haapalainen et al., 2010; Iqbal et al., 2004); excitement (Muldner et al., 2010);
Eyeblink rate	mental workload (Brookings et al., 1996); frustration, stress, anxiety (Kapoor et al., 2007; Doehring, 1957)
Skin-related	
Electro-dermal activity (EDA)	valence, arousal, engagement (Haag et al., 2004; McDuff et al., 2012); frustration (Freeman, 1940; Kapoor et al., 2007); stress and cognitive load (Setz et al., 2010)
Skin temperature	task difficulty (Anthony et al., 2011); valence, arousal (Haag et al., 2004); boredom, engagement, anxiety (Chanel et al., 2008);
Brain-related	
EEG frequency bands	mental workload (Brookings et al., 1996); valence, arousal (Sammler et al., 2007; Reuderink et al., 2013; Lin et al., 2010); happiness and sadness (Li and Lu, 2009); task engagement (Kramer, 1991)
Heart-related	
Heart rate (HR)	mental load & effort (Richter et al., 1998; Veltman and Gaillard, 1998); valence, arousal (Haag et al., 2004; Sammler et al., 2007); positive / negative affect (Drachen et al., 2010); happiness (Steptoe et al., 2005)
Heart rate variability (HRV)	mental effort (Veltman and Gaillard, 1998); task difficulty (Walter and Porges, 1976); anxiety (Rani et al., 2004); various emotional states (McCraty and Tomasino, 2006)
Blood volume pulse (BVP)	cognitive load (Peper et al., 2007); various emotions (Picard et al., 2001); valence, arousal (Haag et al., 2004);
Breathing-related	
Respiratory rate	mental effort (Veltman and Gaillard, 1998); task difficulty (Kuznetsov et al., 2011)

is looking at. This means that to some extent, knowing what a person is looking at gives hints about what they are thinking about.

Eye gaze is commonly measured by an infrared camera that detects reflections on the outermost surface of the eye – so-called corneal reflections. Automatic analysis of the reflection allows the eye tracking device to determine which direction the eye is looking in, to calculate the angle of the eye, and, with the correct calibration, to determine what the eye is looking at.

Eye gaze may be used to calculate *gaze points*, which are the individual samples of what the eye looks at, and which form the base unit of eye gaze measures. If the gaze is held for a long enough duration, this can be interpreted as a *fixation* where the eyes are locked on towards a specific object – giving indications of things like attention and the time required to process what is seen. While fixations are the periods of time when the eye is fixated on a specific object, *saccades* refer to the quick moving of the eye gaze between fixations. Gaze point data may be combined to yield information on the eye gaze sequences used when performing a task. A correctly designed study protocol can use such sequences to infer how properties of the task influence what visual information the study participant focuses on and how she processes it – for example, in terms of how long the person looks at different parts of the visual stimuli.

Pupillary responses or changes in pupil size are caused by two muscles in the eye. One of these, the contracting muscle, receives input from brain systems that respond to light. However, both muscles also receive inputs from areas that are involved in cognitive and autonomic functions. This means that cognitive and autonomic activities influence pupil diameter, and it is possible to gain information on these processes by measuring the pupil. In simple terms, mental exertion leads to dilatation of the pupil, and so the pupil response can be used to infer the level of mental effort.

Pupillary response is commonly measured by an infrared camera that takes into account the distance to the person's eyes. Pupil tracking is already necessary to obtain eye gaze information, and the same algorithms that detect and track the pupil in the overall image are used here.

Eyeblink rate, meaning the frequency at which the eyelids spontaneously open and close, is an indirect measure of dopamine activity in the central nervous system. Dopamine is an important neurotransmitter that is involved in many cognitive and affective functions, such as learning, working memory, and goal-oriented behaviour. The eyeblink rate thus provides clues on what is going on in a person's mind in terms of controlling impulses, maintaining long-term goals, and flexibly adapting to changing rules in the environment.

When the eye is closed, an eye tracker cannot determine the eye gaze or pupillary response. Modern eye trackers include functionality that allows to distinguish eye blinks from other kinds of data loss – such as head movements that prevent the tracker from detecting the eyes.

Eye tracking has several advantages as a research instrument. Sampling rates of commercially available eye tracking devices range from 25 to 2000 measurements per second, meaning that very accurate timing information is available. Eye tracking devices can be completely non-invasive, being attached to a computer monitor, or minimally invasive, in the form of special eye glasses worn by study participants. This means that participants can be observed in quite natural environments and can freely move around. The portability of the devices also means that they can be brought to participants in their own environment rather than having participants visit a lab. Although eye tracking devices have become much more sophisticated in recent years, eye tracking data must always be carefully examined to rule out errors due to changing light conditions, miscalibration, and other sources of errors.

3.1.2 Skin-related Measurements

Commonly used skin-related measurements are skin temperature and electro-dermal activity (EDA). We will focus on EDA in the following.

Electrodermal activity (EDA) is a property of the human skin that causes a continuous variation in its electrical characteristics. More specifically, the ability of the skin to conduct electricity varies, and is linked to the level of psychological or physiological arousal of a person. By measuring electrical properties such as resistance, it is possible to infer, e.g., the level of stress experienced by a person. Since EDA is not under voluntary control, it offers a degree of direct insight into the autonomous regulation of emotions. However, measuring EDA at different locations of the body yields different results, and EDA responses are delayed 1-3 seconds, meaning that it is not straightforward to determine mental activity from an EDA signal. The EDA signal can generally be split into two parts: the slowly changing, low frequency, tonic part, and the fast adapting, high frequency, phasic part (Schmidth and Walach, 2000). Commonly used metrics for the tonic part are the mean value or the area under the curve (AUC), while commonly used features for the phasic part are related to the peaks in the signal.

3.1.3 Brain-related Measurements

Measuring activity of the brain can be done in various ways and the measurements that can be captured depend heavily on the device that is being used. These devices vary in the kind, accuracy and granularity of the data they capture, but also in their invasiveness which limits the kinds of studies that can be performed.

Electroencephalography (EEG) is a method to record electrical activity in the brain. Electrodes placed on the scalp of a participant measure voltage fluctuations that reflect neural activity. When the electrodes are placed correctly, the multiple signals from the many electrodes can be analysed to provide information on activity in different brain regions. This information may be used to investigate cognitive processes. Commonly used measures retrieved from an EEG are brain wave frequency bands that are called alpha (α), beta (β), gamma (γ), delta (δ), and theta (θ). Each of these brain wave frequency bands has a specific frequency range and amplitude and exhibits more or less activity under different stimuli. For example, alpha waves can typically be observed when an individual is in a relaxed state, but the alpha waves either disappear or their amplitude decreases significantly as soon as the physical or mental activity increases (Andreassi, 2007).

EEG has a low spatial resolution, meaning that it can only provide rough information on activations in different brain regions. However, it has high temporal resolution, is less invasive than other sensor technologies, and can be quite mobile and therefore used in situations where participants are moving.

Functional magnetic resonance imaging (fMRI) provides information on the brain activity by measuring changes in blood flow in the brain. When an area of the brain is in active use, blood flow in that area increases. Using fMRI, it is possible to investigate which areas are activated by a stimulus. This may be used to gain knowledge of how participants process the stimulus.

fMRI requires sophisticated equipment and fMRI devices are large and not portable, meaning that participants must visit a research lab to take part in studies. In addition, fMRI generally requires the person to lay still without moving much, and the display that study participants in an fMRI study can look at is also quite small, which limits the kinds of tasks that can be studied with an fMRI significantly.

Functional near-infrared spectroscopy (fNIRS) uses electrical signals close to the infrared wavelengths to detect the composition of materials. As a biosensor, fNIRS works by detecting oxygen saturation in brain tissue. As with fMRI, when a brain region is in active use, blood flow in that region increases, and fNIRS detects this as increased oxygen saturation. A fNIRS device provides more mobility than an fMRI device, but while fMRI can monitor activity in the entire brain, fNIRS is limited to the cortical areas.

3.1.4 Heart-related Measurements

Commonly used heart-related measurements are the *heart rate (HR)*, the *heart rate variability (HRV)*, and the *blood volume pulse (BVP)*. The heart rate refers to the number of contractions of the heart each minute and the heart rate variability represents the variation in the time interval between two consecutive heart beats. The blood volume pulse measures the blood flow through specific parts of the body and may change when the sympathetic nervous system increases its activity, for instance because of stress (Andreassi, 2007). Common features of these measurements are the mean heart rate, the mean and the standard deviation of the time between two heart beats and features that capture the peaks of the BVP signal.

Today, there are various ways to capture heart-related measurements that again vary in the kind, granularity, and accuracy of the data captured. The devices range from commonly used wrist watches, such as the AppleWatch[1], that capture heart rate using an optical sensor, to chest straps or arm bands, all the way to electro-cardiograms that use specific electrodes. Especially the kind of sensor used and the location of the sensor(s) affect the accuracy of the measurements. For instance, capturing accurate HRV data with a wrist watch using an optical sensor is difficult at best: first the wrist is generally used and moved a lot and second, wrist watches are often not tightly fixated to the wrist such that the optical sensor moves around a lot.

Electrocardiography. An electrocardiogram (EKG) is a recording of the electrical activity of the heart over time. Electrodes placed on the skin measure small electrical

[1] https://www.apple.com/watch/

changes that occur when the heart is beating. EKG data may be used to measure physiological arousal connected to a stimulus. If the stimulus increases heart rate, it may be inferred that they are experiencing an emotion – but to determine which emotion, more information is needed, such a self-report instrument.

3.2 Biometric Sensing

Physiological measurements captured by biometric sensors have the potential to tell us more about the human in the process of developing software and to do so in real-time. Yet, there are several points to be considered before using biometrics for studying or supporting software developers. Some of the most prominent of these points are the choice of the biometric sensor or the physiological measurement, the individual differences in the measurements, the processing of the data, and how to best use them for a study or in the field.

3.2.1 Choosing the right measurement and sensor

Gaining insight into the minds of software developers requires careful selection of research instruments. However, picking a biometric sensor will not automatically result in new insights or practically applicable methods. Everything begins with the formulation of research questions or hypothesis informed by existing research and theory, as with any other kind of empirical research (see for example Chapter 1, Chapter 2, and Chapter 8). Without these, the collected data cannot answer any specific questions and is likely to be useless.

The right measurement must thus be informed by an understanding of psychophysiology. In our example, we can observe that Joe is experiencing a state of heightened stress and sleep deprivation, which likely decreases his performance at work. His body will show physiological signs of stress, which can be measured, e.g., by heart sensors or EDA. This could be the basis of a study that tries to associate physiological measures of stress with software development outcomes such as error rate or problem-solving ability.

There are other factors weighing in on the selection of biometric sensors, including the environment and scenario the sensors should be used for, the invasiveness and comfort of the sensor, privacy concerns with the collected data, the required accuracy and granularity of the measurements, and the accessibility of the data and sensor. First, the ***environment and scenario*** that the sensors should be used in limits the choices of sensors. For example, if the study set-up requires an authentic workplace environment the mobility of the sensors plays an important role. So while an fMRI sensor is perfectly reasonable for running precise lab studies and provides very fine-granular data, it is not possible to be used in a realistic setting within open office of a software development company. An eye tracker attached to a computer monitor, on the other hand, can be easily installed even in the workplace without restricting the developer's work, but will also only capture when the developer looks at the screen and not when she might be sketching out some design ideas on a sheet of paper.

Second, and closely related to the first is the ***invasiveness and comfort*** of the sensor. Sensors placed on the fingers might disturb the normal use of a keyboard and mouse, and may be unsuitable for many software development study tasks. EEG sensors will not disturb the normal use of a keyboard and mouse, but usually require time to set-up, be properly placed, might require the user to not walk around too much and can also cause discomfort to the participants, which is a problem both from an ethical perspective and in terms of potentially biasing the study results.

Third, ***privacy concerns*** should be taken into account – biometric data is considered to be among the most sensitive types of data about a person, and steps should be taken to carefully protect study participants. Responsible researchers should think ahead about whether their research leads to greater insights and beneficial applications, or if there is a possibility of misusing the data or results for purposes that put humans at risk.

Fourth, the ***accuracy and granularity*** of the measurements plays an important role. While there is already a large number of sensors to capture a variety of physiological measurements, many of them might not provide the accuracy and granularity that is required for the specific research question under investigation and that was linked to cognitive states and processes in previous research. For instance, there was and is a number of wrist bands to capture heart-related measures that use an optical sensor. However, due to the often loose placement of the wristband and the sensor, the captured data can be very noisy and while that might be sufficient for measuring heart rate, it is often not accurate enough to measure heart rate variability.

Finally, ***accessibility*** of the data provided by the sensor technology also affects the choice of the sensor. For instance, while some biometric sensors might capture the data with the required granularity and sampling rate, the data that researchers have access to can be more limited in terms of the granularity or also the timeliness. Some of the reasons for this limited data access are the goal to reduce the data that needs to be stored and transferred from the often wireless sensor to another device, or the original purpose for the design of the sensor and the limitations in the provided API. An additional factor to keep in mind when choosing a sensor is the continuous support for these sensors. While the market for these technologies is increasing, we are still in the earlier stages of biometric sensing technologies and not all sensor technologies survive, such as the Microsoft Band, to take a prominent example.

3.2.2 Dealing with noisy data and individual differences

Biometric sensors are by nature ***noisy and prone to data loss***. Most devices do some basic filtering of the raw physical signals, but the digital data collected is not ready for analysis as such. Every sensor is susceptible to calibration errors, missing data, noise, or even the weather (Cacioppo et al., 2007). As far as possible, it makes sense to try to minimise error sources already while collecting data. This might mean placing some restrictions on how the participant can move or reducing the amount of disturbances in the environment.

Having the raw digital data potentially poses a need for noise reduction. Often, the biometric sensor manufacturer has instructions and recommendations for how to obtain a cleaner signal. This may include a correction factor that is provided with the digital output of the device itself, or there may be averaging or algorithmic filtering approaches that should be used.

Physiological measures are also inherently ***individual***. While this characteristic of physiological measures allows us to measure and focus on the individual, we also have to pay more attention when comparing data across individuals since the physiological response to certain stimuli can vary significantly across individuals. Let us take heart rate as an example: an individual that is athletic and does sports several times a week most likely has a lower base heart rate than an individual who is sedentary most of the week and additionally the range of the heart rate values will be different. In this case, we cannot directly compare the heart rate with each other. Instead, we have to independently capture a baseline for each individual's heart rate, and then for the periods of interest calculate the difference between the individual's heart rate and her baseline. However, even that is not enough since the range of the heart rate values can vary a lot and we often have to capture the range of individual's heart rate (at least for the states of interest) and then use this to normalize the data to better compare it between individuals. Also, when we measure individuals on a controlled and short task, we might have to take into account and control for certain characteristics, such as their daily rhythm and the participant being a morning or evening person (Levandovski et al., 2013), that can have an impact on the captured measures.Finally, for certain individuals it might not be possible to capture certain physiological measures.

3.3 Analyzing biometric data

Since a lot of biometric data comes in the form of time-series data, appropriate time-series analysis methods should be used. Here, the study design will inform the selection and application of analysis methods. In the following, we will discuss some of the usual steps, however, the concrete steps and order depends on the usage scenario of the biometric data.

One step performed frequently in addition to the cleaning of the data, is ***data segmentation and labelling***. In case of an explicit stimuli in the study design, one often requires an external timing source, such as a video or screen recording, or an automatic signal, that can be used to determine when the stimuli has been displayed, changed, or appeared during the study. This way, there is an objective anchor point in time that can be used to segment and label the data, and that allows to compare the biometric data with and without (or with a different) stimuli. If the study design does not rely on synchronised presentation of a stimuli, other means of analysis are necessary. One approach that is often used is to collect self-reports from individuals for certain points in time or time periods that can then be used for segmentation and data labelling. Another approach is to use objective task data and have participants perform multiple tasks, in which case the task periods can be used for segmentation and the objective measures can be used for labelling.

A further step in the analysis is ***feature extraction***. Generally, we are only interested in specific features of the physiological data. For instance, for heart rate data, we might be interested in the mean and the standard deviation, for EDA data one might be more interested in features related to the peaks in the signal. Previous literature, especially in psychology, can help to determine which feature might be best for a specific use case. In addition, we have to choose specific time windows for extracting and calculating features. The time window choice depends on the kind of physiological measure, the stimuli and other factors and can be quite challenging, especially since the optimal time window for extracting a feature might even vary across participants.

In many cases, biometric data is used to ***build a machine learning model*** that can be used to determine which biometric features are best to classify or predict certain states. For this step, the data has to be split into training and test data and depending on the evaluation, e.g. cross-validation or leave-one-out, different methods for splitting have to be considered. It is thereby important to ensure that the training and test data, and in general the individual data points, do not overlap and are independent of each other. For biometric data, we have to pay special attention to this independence since physiological features of an individual can be affected for a long time by a certain stimuli. Therefore, we might have to consider adding periods of rest or longer breaks in between tasks or time segments to ensure that the effect of the stimuli has worn off. As with non-biometric data, another challenge in training a machine learning classifier for a specific use case is the selection of the features to be used. Depending on the machine learning method applied, different feature selection methods can be chosen. Given the often large number of possible biometric features (also based on the various time windows that can be used to extract the features), it can be challenging to determine an optimal set of features.

Ultimately, using biometric measurements in software engineering requires an understanding of phycho-physiology, operational knowledge of the biometric devices, data analysis skills, skills in study design, and an understanding of the real-life tasks that software developers carry out in their work.

4 Work in the Area

Biometric sensor data has been used in software engineering research for various purposes. Some of this research addresses questions that contribute to an improved understanding of the individual in the process of developing software, while other research explores how biometric sensor data could be used to build new tools and support the developer. Yet other research explores the sensor data itself and tries to understand what type of sensor data or combination thereof works best for detecting certain states of a developer or for studying specific types of tasks. In the following, we will focus on three areas of research: using eye-tracking to understand developer's code comprehension, examining developers' brains with fMRI or EEG, and more general sensing of specific aspects of developers, such as experienced task difficulty or their emotions.

Overall, the research in this area is increasing and is helping to better understand software developers in their activities which can then be used to improve tool design. Furthermore, the real-time measurement and prediction of aspects such as a developer's cognitive load when reading code can be used to build novel tool support, for instance, to predict code difficulty and which code to refactor, to avoid bugs from being committed, and to signal the interruptibility to co-workers and avoid expensive interruptions.

4.1 Tracking developers' eyes in the code

Some long-standing questions in software engineering research have to do with source code and how developers read and comprehend source code. Since developers spend significant amounts of tie on reading and writing source code, various methods have been used to study developers during these activities, and more recently with the help of eye tracking.

Several studies have used eye tracking to examine how developers read algorithms and source code, what they focus on most when reading code snippets and how the reading of code compares to reading natural text, e.g. (Crosby and Stelovsky, 1990; Busjahn et al., 2015; Rodeghero et al., 2014; Uwano et al., 2006). One of the interesting findings is that reading natural text happens largely linearly, in western languages, left-to-right and top-to-bottom. For source code, eye tracking has revealed that the reading is different and that experts read source code in a less linear manner than novices (Busjahn et al., 2015). Another finding is that initial code segments (e.g., in a function) are read more times and receive more focus, while later parts may only be skimmed (Jbara and Feitelson, 2017).

Eye tracking has also been used to investigate how developers perform change tasks (Kevic et al., 2015), code reviews (Chandrika et al., 2017; Begel and Vrzakova, 2018), and to trace requirements through the software lifecycle (Sharif et al., 2017). These examples show the great diversity of research topics that may be addressed with eye tracking.

4.2 Examining developers' brains

Recently, researchers have started to look deeper into the brain using fMRI to investigate program comprehension. For example, in a study using fMRI, researchers found that that program comprehension was associated with a specific activation pattern in five brain regions related to working memory, attention, and language processing (Siegmund et al., 2014; Peitek et al., 2018). In another fMRI study, Siegmund et al. found evidence of semantic chunking during bottom-up code comprehension – when a developer has to interpret every individual program statement to form an understanding of the program – and found that semantic cues, such as method signatures and common programming idioms, ease comprehension (Siegmund et al., 2017). Floyd et al. further used fMRI to examine the difference between reviewing code and English prose and found that the neural representations of programming and natural language are distinct and that they are affected by expertise (Floyd et al., 2017).

The high spatial resolution of fMRI allows researchers to pinpoint precise areas that are activated in the brain. In contrast, EEG measurements are less invasive to collect and have lower spatial resolution but much higher temporal resolution, allowing studies to be more precise in terms of timing. In one EEG study, it was shown that brain measures of cognitive load could quantify programming experience among students – in terms of the state of progression through an undergraduate computer science program – and self-reported experience level (Crk et al., 2015). Code comprehension has also been investigated using EEG. A study identified EEG signatures specific to code comprehension, and found neural correlates of subjective difficulty during code comprehension by using study participants' ratings of the tasks they had to perform in the study (Kosti et al., 2018).

4.3 Sensing developers

One of the earliest approaches in the software engineering domain mentioning biometric sensing is the Ginger2 environment by Torii and colleagues that talked about the use of an eye-tracking and a skin resistance level sensor to empirically study developers (Torii et al., 1999). Since then, there is an increasing amount of studies that examines further aspects of developers and/or combine various physiological measures and sensors. Several studies focused on the ***difficulty*** of the task or the cognitive load that developers experience during a change task or when reviewing or comprehending small code snippets. One study, for instance, examined the use of various physiological measurements to predict difficulty of small code snippets (Fritz et al., 2014). In another study on short code comprehension and bug localization tasks, the researchers used a combination of fNIRS and eye tracking and found that linguistic antipatterns – poor practices in naming, documentation and choice of identifiers – in the source code significantly increased the developers' cognitive load (Fakhoury et al., 2018). In a longer term study conducted over the course of a week, a combination of heart-, breathing, and skin-related measurements from two different sensors were captured in combination with computer interaction data to predict the difficulty of code elements and the quality of the code produced. Using a machine learning approach, the different biometric readings were used to construct a prediction model that was able to outperform a model based only on traditional (non-biometric) measurements (Müller and Fritz, 2016).

Research has also looked into the use of biometric sensors to investigate developers' ***emotions***. These studies can also benefit from using multiple sensors, since emotions manifest in several physiological responses. A study combining non-invasive, low-cost EEG, EMG, and GSR (EDA) sensors found that it was possible to obtain accurate classification of emotional valence and arousal using machine learning classifiers on the sensor data (Girardi et al., 2017). A study using a variety of biosensor data (EEG, EDA, skin temperature, heart rate, blood volume pulse, eye tracking) to build a machine learning model found that it was possible to distinguish between positive and negative emotions and low and high progress during software change tasks (Müller and Fritz, 2015). In general though, predicting arousal – the amount of activation associated with an emotion – is easier than predicting valence –the positive or negative character of an emotion – as several studies in other fields have also shown.

Finally, studies have also investigated other aspects, such as interruptibility – the availability of a developer for an interruption. One study, for instance, investigated how biometric and interaction data could be used to predict interruptibility in an office setting (Züger et al., 2018). Computer interaction data was shown to be more accurate than biometric data alone, but a combination of both yielded the best results.

5 Recommended Further Reading

The literature on using biometric measurements in software engineering is growing, and a few years from now, we expect there to be a large body of research on the subject. However, there is already a larger body of literature in areas such as psychology or human computer interaction, in which physiological measures have been explored in a variety of settings and tasks, such as car driving, physical exercise, specific cognitive skills tasks, and more.

In the previous sections, we have already listed quite a few of the relevant articles and books that can help you to gain a better understanding on the different types of physiological measures and which cognitive and emotional states they have been linked to especially based on previous work in psychology (see Section 3), but also on the types of studies that have been conducted in software engineering using biometrics (see Section 4). Many of the referenced literature as well as the research on biometrics in human computer interaction and psychology, can be a valuable source for understanding which measures best to use for which scenario, the challenges involved in the use of certain biometric sensors, how to analyze the data and how to design studies, and sometimes even provide the code for the analysis.

There is a range of further readings that, depending on the type of study and the biometric sensor(s) to be used, can be of relevance. For instance, for eye-tracking there are several articles by Sharif et al. on the use of eye tracking metrics and the mapping of eye gazes to areas of interest in the code, as well as a tool that can help with the mapping (Sharif and Shaffer, 2015; Sharif et al., 2016). Other papers look at and compare the use of low-cost EEG devices (e.g., (Das et al., 2014)), discuss the invasiveness of various sensors to detect emotions (e.g., (Wrobel, 2018)), or also summarize some of the work in the area of using biometrics to increase developer productivity (Fritz and Müller, 2016).

Due to the cross-disciplinary nature of this type of research, literature relevant for research in the area is not confined to the software engineering domain. There are various other domains that are relevant, such as psychology, human computer interaction, but also conference proceedings or journals about sensor technology and more. We therefore strongly recommend that in addition to the software engineering literature, you also look outside the field, since there is much to be learnt from previous studies. Finally, sensor technology is rapidly changing and the companies that provide these technologies as well, so it is worthwhile to regularly explore what kind of sensors are available to measure certain physiological features.

6 Conclusion

Biometric or psycho-physiological measurement is an emerging and promising source of information that can help researchers and practitioners to better understand and support developers in their work. Biometric sensors provide a direct way of measuring physiological correlates of psychological processes that are active as developers conduct their work. When used correctly, biometric data can yield information that is not possible to obtain using other means and that allows us to capture more about an individual in real-time.

So far, research using biometric sensor data in the software engineering domain is in its early stages, yet there is a huge potential as previous work has already shown. Research in the area ranges from the use of eye tracking to examine code comprehension, to the use of fMRIs to detect brain activation patterns for code reviewing tasks, all the way to the use of less invasive heart-related sensors in the field to detect code difficulty and the likelihood of a developer to create a bug. Initial results in the area already demonstrate the potential that biometric data has for measuring a developer's cognitive and emotional states in real-time, however, there are also still several challenges to be addressed in the future.

By leveraging biometric data in the software engineering domain and by having real-time measures of a developers' cognitive and emotional processes during work, there are many new opportunities that open up to better understand a developer in the process and to train and support developers for and in their work. The possible developer support ranges from preventing developers from creating or committing a bug, to detecting difficult areas in a code base (areas with high technical debt), to helping software developers to stay focused and take breaks, regulating the amount of information available to the developer at a given point in time and avoiding stress, to actually assisting the developer in their decision-making tasks as well as in signalling to co-workers when a developer is available for an interruption or not. Future software development tools could be collaborative agents that are informed not only by models of the system being developed, but also by information on the developers and potentially other stakeholders and thus provide better and more tailored support. Especially with the fast advances in sensor and data analysis technology, we might soon all be wearing smart wearable devices with biometric sensors integrated that will already be accurate enough to provide some of this support.

To understand software development, you must understand software developers. Biometric measurements have the potential to significantly help us a lot in this regard and thereby not just change our understanding but also our way of developing software.

References

Andreassi JL (2007) Psychophysiology: Human Behavior & Physiological Response. Lawrence Erlbaum Associates

Anthony L, Carrington P, Chu P, Kidd C, Lai J, Sears A (2011) Gesture dynamics: Features sensitive to task difficulty and correlated with physiological sensors. Stress 1418(360)

Ayres P (2001) Systematic mathematical errors and cognitive load. In: Contemporary Educational Psychology

Bailey BP, Konstan JA, Carlis JV (2001) The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: Proceedings of INTERACT, vol 1, pp 593–601

Bednarik R, Tukiainen M (2006) An eye-tracking methodology for characterizing program comprehension processes. In: Proceedings of the Symposium on Eye Tracking Research & Applications, pp 125–132

Begel A, Vrzakova H (2018) Eye movements in code review. In: Proceedings of the Workshop on Eye Movements in Programming, ACM, New York, NY, USA, EMIP '18, pp 5:1–5:5, DOI 10.1145/3216723.3216727

Brookings JB, Wilson GF, Swain CR (1996) Psychophysiological responses to changes in workload during simulated air traffic control. Biological Psychology: Psychophysiology of Workload 42(3):361 – 377

Busjahn T, Bednarik R, Begel A, Crosby M, Paterson JH, Schulte C, Sharif B, Tamm S (2015) Eye movements in code reading: Relaxing the linear order. In: 2015 IEEE 23rd International Conference on Program Comprehension, pp 255–265, DOI 10.1109/ICPC.2015.36

Cacioppo J, Tassinary LG, Berntson GG (2007) The Handbook of Psychophysiology. Cambridge

Carniglia E, Caputi M, Manfredi V, Zambarbieri D, Pessa E (2012) The influence of emotional picture thematic content on exploratory eye movements. Journal of Eye Movement Research 5(4):1–9

Chandrika KR, Amudha J, Sudarsan SD (2017) Recognizing eye tracking traits for source code review. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp 1–8, DOI 10.1109/ETFA.2017.8247637

Chanel G, Rebetez C, Bétrancourt M, Pun T (2008) Boredom, engagement and anxiety as indicators for adaptation to difficulty in games. In: Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era, pp 13–17, DOI 10.1145/1457199.1457203, URL http://doi.acm.org/10.1145/1457199.1457203

Crk I, Kluthe T, Stefik A (2015) Understanding programming expertise: An empirical study of phasic brain wave changes. ACM Trans Comput-Hum Interact 23(1):2:1–2:29, DOI 10.1145/2829945

Crosby ME, Stelovsky J (1990) How do we read algorithms? a case study. Computer 23(1):25–35, DOI 10.1109/2.48797

Czerwinski M, Cutrell E, Horvitz E (2000) Instant messaging: Effects of relevance and timing. In: People and computers XIV: Proceedings of HCI, British Computer Society, vol 2, pp 71–76

Das R, Chatterjee D, Das D, Sinharay A, Sinha A (2014) Cognitive load measurement - a methodology to compare low cost commercial eeg devices. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp 1188–1194, DOI 10.1109/ICACCI.2014.6968528

Doehring DG (1957) The relation between manifest anxiety and rate of eyeblink in a stress situation. Tech. rep., Central Institute for the Deaf, St Louis, MO

Drachen A, Nacke LE, Yannakakis G, Pedersen AL (2010) Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. In: Proceedings of the 5th Symposium on Video Games, pp 49–54

Fakhoury S, Ma Y, Arnaoudova V, Adesope O (2018) The effect of poor source code lexicon and readability on developers' cognitive load. In: Proceedings of the 26th Conference on Program Comprehension, ACM, New York, NY, USA, ICPC '18, pp 286–296, DOI 10.1145/3196321.3196347

Floyd B, Santander T, Weimer W (2017) Decoding the representation of code in the brain: An fmri study of code review and expertise. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp 175–186, DOI 10.1109/ICSE.2017.24

Freeman GL (1940) A method of inducing frustration in human subjects and its influence upon palmar skin resistance. The American Journal of Psychology 53(1):pp. 117–120

Fritz T, Müller SC (2016) Leveraging biometric data to boost software developer productivity. In: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), vol 5, pp 66–77

Fritz T, Begel A, Müller SC, Yigit-Elliott S, Züger M (2014) Using psychophysiological measures to assess task difficulty in software development. In: Proceedings of the 36th International Conference on Software Engineering, ACM, ICSE 2014, pp 402–413, DOI 10.1145/2568225.2568266

Girardi D, Lanubile F, Novielli N (2017) Emotion detection using noninvasive low cost sensors. In: 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), pp 125–130, DOI 10.1109/ACII.2017.8273589

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, DOI 10.1016/j.jss.2018.02.041

Haag A, Goronzy S, Schaich P, Williams J (2004) Emotion recognition using bio-sensors: First steps towards an automatic system. Affective Dialogue Systems Lecture Notes in Computer Science 3068:36–48

Haapalainen E, Kim S, Forlizzi JF, Dey AK (2010) Psycho-physiological measures for assessing cognitive load. In: Proceedings of the 12th International Conference on Ubiquitous Computing, pp 301–310

Ikehara CS, Crosby ME (2005) Assessing cognitive load with physiological sensors. In: Proceedings of the 38th Hawaii International Conference on System Sciences, p 295a

Ikutani Y, Uwano H (2014) Brain activity measurement during program comprehension with NIRS. In: Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp 1–6

Iqbal ST, Zheng XS, Bailey BP (2004) Task-evoked pupillary response to mental workload in human-computer interaction. In: CHI '04 Extended Abstracts on Human Factors in Computing Systems, pp 1477–1480

Jbara A, Feitelson DG (2017) How programmers read regular code: a controlled experiment using eye tracking. Empirical Software Engineering 22(3):1440–1477, DOI 10.1007/s10664-016-9477-x

Kapoor A, Burleson W, Picard RW (2007) Automatic prediction of frustration. International Journal of Human-Computer Studies 65(8):724–736

Kevic K, Walters BM, Shaffer TR, Sharif B, Shepherd DC, Fritz T (2015) Tracing software developers' eyes and interactions for change tasks. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ACM, New York, NY, USA, ESEC/FSE 2015, pp 202–213, DOI 10.1145/2786805.2786864

Kosti MV, Georgiadis K, Adamos DA, Laskaris N, Spinellis D, Angelis L (2018) Towards an affordable brain computer interface for the assessment of programmers' mental workload. International Journal of Human-Computer Studies 115:52–66, DOI 10.1016/j.ijhcs.2018.03.002

Kramer AF (1991) Physiological metrics of mental workload: A review of recent progress. Multiple-task Performance pp 279–328

Kuznetsov NA, Shockley KD, Richardson MJ, Riley MA (2011) Effect of precision aiming on respiration and postural-respiratory synergy. Neuroscience letters 502(1):13–17

Levandovski R, Sasso E, Hidalgo MP (2013) Chronotype: a review of the advances, limits and applicability of the main instruments used in the literature to assess human phenotype. Trends in psychiatry and psychotherapy 35(1):3–11

Li M, Lu BL (2009) Emotion classification based on gamma-band EEG. Conference Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society pp 1323–1326

Lin YP, Wang CH, Jung TP, Wu TL, Jeng SK, Duann JR, Chen JH (2010) Eeg-based emotion recognition in music listening. IEEE Transactions on Biomedical Engineering 57(7):1798–1806, DOI 10.1109/TBME.2010.2048568

Mark G, Gudith D, Klocke U (2008) The cost of interrupted work: more speed and stress. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, ACM, pp 107–110

McCraty R, Tomasino D (2006) Stress in Health and Deseases, Wiley-VCH, chap Emotional Stress, Positive Emotions, and Psychophysiological Coherence

McDuff D, Karlson A, Kapoor A, Roseway A, Czerwinski M (2012) Affectaura: an intelligent system for emotional memory. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, pp 849–858

Muldner K, Burleson W, VanLehn K (2010) "Yes!": using tutor and sensor data to predict moments of delight during instructional activities. In: Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization, pp 159–170

Müller SC, Fritz T (2015) Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress. In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol 1, pp 688–699, DOI 10.1109/ICSE.2015.334

Müller SC, Fritz T (2016) Using (bio)metrics to predict code quality online. In: 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp 452–463, DOI 10.1145/2884781.2884803

Nakagawa T, Kamei Y, Uwano H, Monden A, Matsumoto K, German DM (2014) Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: A controlled experiment. In: Companion Proceedings of International Conference on Software Engineering

Peitek N, Siegmund J, Apel S, Kästner C, Parnin C, Bethmann A, Leich T, Saake G, Brechmann A (2018) A look into programmers' heads. IEEE Transactions on Software Engineering DOI 10.1109/TSE.2018.2863303

Peper E, Harvey R, Lin IM, Tylova H, Moss D (2007) Is there more to blood volume pulse than heart rate variability, respiratory sinus arrhythmia, and cardiorespiratory synchrony? Biofeedback 35(2)

Picard RW, Vyzas E, Healey J (2001) Toward machine emotional intelligence: Analysis of affective physiological state. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10):1175–1191

Rani P, Sarkar N, Smith CA, Kirby LD (2004) Anxiety detecting robotic system - towards implicit human-robot collaboration. Robotica 22(1):85–95

Reuderink B, Mühl C, Poel M (2013) Valence, arousal and dominance in the eeg during game play. International Journal of Autonomous and Adaptive Communications Systems 6(1):45–62

Richter P, Wagner T, Heger R, Weise G (1998) Psychophysiological analysis of mental load during driving on rural roads - a quasi-experimental field study. Ergonomics 41(5)

Rodeghero P, McMillan C, McBurney PW, Bosch N, D'Mello S (2014) Improving automated source code summarization via an eye-tracking study of programmers. In: Proceedings of the 36th International Conference on Software Engineering, ICSE 2014, pp 390–401, DOI 10.1145/2568225.2568247

Sammler D, Grigutsch M, Fritz T, Koelsch S (2007) Music and emotion: electrophysiological correlates of the processing of pleasant and unpleasant music. Psychophysiology 44:293–304, DOI 10.1111/j.1469-8986.2007.00497.x

Schmidth S, Walach H (2000) Electrodermal activity (EDA) - state-of-the-art measurements and techniques for parapsychological purposes. Journal of Parapsychology 64(2)

Setz C, Arnrich B, Schumm J, Marca RL, Tröster G, Ehlert U (2010) Discriminating stress from cognitive load using a wearable eda device. Trans on Information Technology in Biomedicine 14(2)

Sharif B, Shaffer T (2015) The use of eye tracking in software development. In: Schmorrow DD, Fidopiastis CM (eds) Foundations of Augmented Cognition, Springer International Publishing, Cham, pp 807–816

Sharif B, Falcone M, Maletic JI (2012) An eye-tracking study on the role of scan time in finding source code defects. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, ETRA '12, pp 381–384, DOI 10.1145/2168556.2168642

Sharif B, Clark B, Maletic JI (2016) Studying developer gaze to empower software engineering research and practice. In: Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ACM, New York, NY, USA, FSE 2016, pp 940–943, DOI 10.1145/2950290.2983988

Sharif B, Meinken J, Shaffer T, Kagdi H (2017) Eye movements in software traceability link recovery. Empirical Software Engineering 22(3):1063–1102, DOI 10.1007/s10664-016-9486-9

Siegmund J, Kästner C, Apel S, Parnin C, Bethmann A, Leich T, Saake G, Brechmann A (2014) Understanding understanding source code with functional magnetic resonance imaging. In: Proceedings of the 36th International Conference on Software Engineering, pp 378–389

Siegmund J, Peitek N, Parnin C, Apel S, Hofmeister J, Kästner C, Begel A, Bethmann A, Brechmann A (2017) Measuring neural efficiency of program comprehension. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ACM, New York, NY, USA, ESEC/FSE 2017, pp 140–150, DOI 10.1145/3106237.3106268

Steptoe A, Wardle J, Marmot M (2005) Positive affect and health-related neuroendocrine, cardiovascular, and inflammatory processes. Proceedings of the National Academy of Sciences 102(18):6508–6512

Sweller J (1988) Cognitive load during problem solving: Effects on learning. Cognitive Science 12(2):257–285

Torii K, Matsumoto Ki, Nakakoji K, Takada Y, Takada S, Shima K (1999) Ginger2: An environment for computer-aided empirical software engineering. Software Engineering, IEEE Transactions on 25(4):474–492

Uwano H, Nakamura M, Monden A, Matsumoto Ki (2006) Analyzing individual performance of source code review using reviewers' eye movement. In: Proceedings of the Symposium on Eye Tracking Research & Applications, ACM, San Diego, California, pp 133–140

Veltman J, Gaillard AW (1998) Physiological workload reactions to increasing levels of task difficulty. Ergonomics 41(5):656–669

Walter GF, Porges SW (1976) Heart rate and respiratory responses as a function of task difficulty: The use of discriminant analysis in the selection of psychologically sensitive physiological responses. Psychophysiology 13(6)

Wilson GF (1992) Applied use of cardiac and respiration measures: Practical considerations and precautions. Biological Psychology 34(2–3):163–178

Wrobel MR (2018) Applicability of emotion recognition and induction methods to study the behavior of programmers. Applied Sciences 8(323), DOI 10.3390/app8030323

Züger M, Müller SC, Meyer AN, Fritz T (2018) Sensing interruptibility in the office: A field study on the use of biometric and computer interaction sensors. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, CHI '18, pp 591:1–591:14, DOI 10.1145/3173574.3174165