Assessing Cognitive Load in Software Development with Wearable Sensors

1st Fabian Stolp
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
fabian.stolp@hpi.de

***Abstract*—The understandability of source code influences software quality, and being able to measure it could greatly benefit software development and maintenance. There is an ongoing debate about the validity of using software metrics for this purpose. In this context, software developers' cognitive load during code comprehension is increasingly often investigated. The concept of cognitive load provides information about the usage of mental resources. Previous research has shown that cognitive load is derivable from physiological measurements. This paper proposes using wearable body sensors that can be easily deployed in software development settings and for empirical software engineering research to provide a cognitive perspective on code understandability and software metrics.**

***Index Terms*—program comprehension, code understandability, cognitive load, wearable sensors, software metrics**

I. Motivation

Today, software permeates our societies, and software defects can harm every aspect of our lives, sometimes even costing lives. Looking at malfunctioning software's causes, we must consider human factors during its development. One of the major tasks software developers spend time on is code comprehension [1], [2]. It has been shown that the comprehensibility of code influences, among others, software quality [3] and the work performance of developers [4]. Methods like think-aloud protocols, interviews, or behavioral measures can be used to evaluate how well developers understand code [4], [5]. However, each method has its downsides, such as the possibility of introducing biases during interviews. A direct, objective way of comprehensibility assessment would be valuable.

For the quantification of software properties, software metrics can be used, and a subset has been developed to evaluate the comprehensibility of source code. Others are at least often assumed to evaluate this property [6]. Overall, software metrics should provide values that enable decisions, for example, regarding the need for refactoring. However, the validity and validation of those metrics are continuously discussed in the literature [7]–[10]. For metrics that are thought to provide information about the understandability of code, it seems logical to consider a developer-centered approach and examine a metric's validity from a cognitive viewpoint.

That cognitive processes play an essential role in software engineering has been discussed from diverse perspectives [5], [7]. An overview of cognitive concepts found in the literature in the context of software engineering provide Fagerholm et al. in their recent literature review [5]. One of these concepts is cognitive load. Roughly speaking, the cognitive load a person experiences can provide hints about how many mental resources they have to allocate to process the information at hand [11]. High cognitive load is associated with decreased performance and an increased error rate [4], [12]. Thus, it seems promising to consider this concept when evaluating the understandability of source code and the validity of corresponding software metrics.

Various studies have shown that cognitive load can be derived from physiological measurements [13]–[15]. Physiological parameters influenced by the autonomous nervous system, such as heart rate (variability), electrodermal activity [15], and eye/pupil movements [14], allow conclusions to be drawn about the cognitive load experienced. Even brain imaging measurements can be considered. For example, the brain's electrical activity that can be measured by electroencephalography (EEG) was evaluated in cognitive load studies [13]. However, factors such as noise can lead to limitations when considering only single modalities. Compensating such factors by taking a multimodal approach and fusing the data from different sensors can yield more robust results [16], [17].

In this context, it is relevant to consider that wearable body sensors are getting increasingly cheaper and more convenient. That is true for a variety of devices. The trend even applies to wearables with brain imaging capabilities. Such increasingly more precise, low-cost devices that are convenient to apply in diverse, realistic environments yield the promise of novel insights, also for empirical software engineering research [18].

Indeed, software engineering activities are increasingly often investigated by analyzing physiological measurements [18], and it has been seen that the cognitive load derivable from physiology and code comprehension success are connected [4], [19]. Recently, studies evaluating the validity of software metrics using such insights were conducted [19]–[21], and the need for further studies was stated [6], [18].

While prices for body sensors are falling and the interest in a

The work of Fabian Stolp is funded by the Hasso Plattner Institute Research School on Data Science and Engineering.

979-8-3503-2263-7/23/$31.00 ©2023 IEEE

physiological evaluation of source code understandability and corresponding metrics is increasing, further data collection, analysis, and frameworks that support corresponding studies are needed [6], [18]. Thus, collecting data using a low-cost multimodal setup of sensors that can be easily applied in software engineering research, evaluating the data, especially considering its quality and the validity of the previously mentioned software metrics, and creating a framework for such a data collection and analysis are important pending challenges. We want to focus our work on EEG, eye tracking, pupillometry, heart rate, and electrodermal activity devices that can be conveniently used in diverse software development settings.

II. Hypotheses

The overarching claim is that (1) A multimodal setup of low-cost wearable devices provides data of a quality that makes it usable for the evaluation of cognitive load during code comprehension. It thus represents a valuable addition to the methods used in empirical software engineering research.

Narrowing down the scope and concentrating on the validity of software metrics, we hypothesize that (2) The validation and design of code complexity metrics can benefit from psychophysiological evaluations of these metrics.

On a finer granularity level, we would like to investigate the following claim: (3) Code metrics that show a higher correlation with physiologically derived cognitive load provide more consistent results than those with a lower correlation regarding the comprehension performance of developers.

III. Contributions and Related Work

A general overview of psychophysiological measures in software engineering research is given in [18]. More specifically, evaluating cognitive load in software engineering settings using wearables without focusing on software metrics has been looked into before [4], [22], [23]. Prior studies also considered the validity of software metrics regarding physiologically derived cognitive load. However, in several of these studies, expensive or stationary devices were used [19], [24]. Recently, some also considered this question by applying less costly body sensors [20], [21]. However, combining brain imaging devices and a set of further, less intrusive physiological sensors was only referred to as future work [20], thus leaving a gap concerning the use of such multimodal setups of sensors in this field. In addition, the devices that were used before require a considerable effort to apply and are less convenient than novel wearables. Last but not least, so far, in this way, only a subset of metrics has been investigated. Overall, data are still limited, and further research, tool support, and data are often requested [6], [18].

Considering the related work and open questions, we want to work on the following contributions. (1) The evaluation of the use and utility of multimodal measurements from wearable brain imaging and less intrusive physiological sensors in empirical software engineering, especially regarding the validity of software metrics. (2) Creating and providing a framework for the low-cost psychophysiological validation of software metrics. (3) The investigation of cognitive processes influencing the comprehension performance of software developers.

Concrete steps have been taken, among others, by preparing a study comparing cognitive load derived from low-cost psychophysiological measurements with subjective perception, behavior, and code complexity metric scores for program comprehension tasks. The University of Potsdam ethics committee approved the corresponding ethics application so that participant recruiting and implementing the study can start.

Devices that will be used include Emotiv Epoc-X EEG devices, Tobii Nano Pro eye trackers, and Empatica E4 wristbands or Shimmer3 GSR+ units. In [19], Peitek et al. evaluated, among others, the correlation of cognitive load derived from functional magnetic resonance imaging (fMRI) and specific software metrics. Together with their results, they also published the materials used. To facilitate the comparability between results that have been achieved by using more expensive, stationary equipment and the results that will be achieved using a multimodal, low-cost, wearable setup, we plan to include the same Java code snippets and the range of code metrics as used in [19]. The open-source software PsychoPy [25] and Lab Streaming Layer (LSL)[1] will be used for functions such as task presentation, data recording, and synchronization. Participation in the study will be advertised through mailing lists and notice boards. Furthermore, we are evaluating the inclusion of development teams from a multinational software company.

IV. Evaluation

The evaluation of results can be looked at on different granularity levels. The analysis of the physiological measures will rely on previous research on cognitive load and physiology. Overviews of corresponding research are, for example, given in [4], [15], [18]. To give an example, for EEG data, the evaluation of the power of frequency bands will be one of the possibilities used to derive cognitive load [13]. Validated questionnaires will be used to assess developer characteristics. For example, for evaluating developer experience, the one by Siegmund et al. will be used [26]. To evaluate the perceived complexity of comprehension tasks, participants will compare code snippets relative to each other. Furthermore, using the NASA Task Load Index [27] is a potential further solid possibility to evaluate the perceived workload. Behavioral measures will comprise the correctness of answers and the time taken to answer questions. In the concrete case of the before mentioned study, such questions would ask for comprehending source code by providing the output of a code snippet given its input.

To bring everything together, we will determine the informativeness of software metrics correlated with cognitive load by calculating correlations with other measures used before to evaluate code comprehensibility and software metrics, like behavioral measures and perceived complexity. Then, by analyzing the results and considering previous work, we will

[1] https://labstreaminglayer.org/

examine the implications for cognitive load measurement in software development and metrics validation and design.

References

[1] I. Schröter, J. Krüger, J. Siegmund, and T. Leich, "Comprehending studies on program comprehension," in *Proceedings of the 25th International Conference on Program Comprehension*, ser. ICPC '17. IEEE Press, 2017, p. 308–311.

[2] J. Siegmund and J. Schumann, "Confounding parameters on program comprehension: a literature survey," *Empirical Software Engineering*, vol. 20, no. 4, pp. 1159–1192, Aug 2015.

[3] A. Schankin, A. Berger, D. V. Holt, J. C. Hofmeister, T. Riedel, and M. Beigl, "Descriptive compound identifier names improve source code comprehension," in *Proceedings of the 26th Conference on Program Comprehension*, ser. ICPC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 31–40.

[4] S. Müller, "Using biometric sensors to increase developers' productivity," Ph.D. dissertation, University of Zurich, Zurich, April 2016.

[5] F. Fagerholm, M. Felderer, D. Fucci, M. Unterkalmsteiner, B. Marculescu, M. Martini, L. G. W. Tengberg, R. Feldt, B. Lehtelä, B. Nagyváradi, and J. Khattak, "Cognition in software engineering: a taxonomy and survey of a half-century of research," *ACM Comput. Surv.*, vol. 54, no. 11s, Sep 2022.

[6] M. Muñoz Barón, M. Wyrich, and S. Wagner, "An empirical validation of cognitive complexity as a measure of source code understandability," in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ser. ESEM '20. New York, NY, USA: Association for Computing Machinery, 2020.

[7] F. Détienne, *Software design–cognitive aspects*, F. Bott, Ed. New York, NY: Springer-Verlag, 2002.

[8] A. Meneely, B. Smith, and L. Williams, "Validating software metrics: a spectrum of philosophies," *ACM Trans. Softw. Eng. Methodol.*, vol. 21, no. 4, Feb 2013.

[9] Y. Gil and G. Lalouche, "On the correlation between size and metric validity," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2585–2611, Oct 2017.

[10] S. Scalabrino, G. Bavota, C. Vendome, M. Linares-Vásquez, D. Poshyvanyk, and R. Oliveto, "Automatically assessing code understandability," *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 595–613, 2021.

[11] F. G. W. C. Paas and J. J. G. Van Merriënboer, "Instructional control of cognitive load in the training of complex cognitive tasks," *Educational Psychology Review*, vol. 6, no. 4, pp. 351–371, Dec 1994.

[12] R. Weast and N. Neiman, "The effect of cognitive load and meaning on selective attention," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 32, 2010.

[13] P. Antonenko, F. Paas, R. Grabner, and T. van Gog, "Using electroencephalography to measure cognitive load," *Educational Psychology Review*, vol. 22, no. 4, pp. 425–438, Dec 2010.

[14] J. Zagermann, U. Pfeil, and H. Reiterer, "Measuring cognitive load using eye tracking technology in visual computing," in *Proceedings of the Sixth Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization*, ser. BELIV '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 78–85.

[15] E. Haapalainen, S. Kim, J. F. Forlizzi, and A. K. Dey, "Psycho-physiological measures for assessing cognitive load," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, ser. UbiComp '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 301–310.

[16] P. Vanneste, A. Raes, J. Morton, K. Bombeke, B. B. Van Acker, C. Larmuseau, F. Depaepe, and W. Van den Noortgate, "Towards measuring cognitive load through multimodal physiological data," *Cognition, Technology & Work*, vol. 23, no. 3, pp. 567–585, Aug 2021.

[17] F. Chen, J. Zhou, Y. Wang, K. Yu, S. Z. Arshad, A. Khawaji, and D. Conway, *Robust Multimodal Cognitive Load Measurement*, 1st ed., ser. Human–Computer Interaction Series. Cham: Springer International Publishing, 2016.

[18] B. Weber, T. Fischer, and R. Riedl, "Brain and autonomic nervous system activity measurement in software engineering: a systematic literature review," *Journal of Systems and Software*, vol. 178, p. 110946, 2021.

[19] N. Peitek, S. Apel, C. Parnin, A. Brechmann, and J. Siegmund, "Program comprehension and code complexity metrics: an fmri study," in *Proceedings of the 43rd International Conference on Software Engineering*, ser. ICSE '21. IEEE Press, 2021, p. 524–536.

[20] J. Medeiros, R. Couceiro, G. Duarte, J. Durães, J. Castelhano, C. Duarte, M. Castelo-Branco, H. Madeira, P. de Carvalho, and C. Teixeira, "Can eeg be adopted as a neuroscience reference for assessing software programmers' cognitive load?" *Sensors*, vol. 21, no. 7, 2021.

[21] G. Hao, H. Hijazi, J. Durães, J. Medeiros, R. Couceiro, C. T. Lam, C. Teixeira, J. Castelhano, M. C. Branco, P. Carvalho, and H. Madeira, "On the accuracy of code complexity metrics: a neuroscience-based guideline for improvement," *Frontiers in Neuroscience*, vol. 16, 2023.

[22] M. V. Kosti, K. Georgiadis, D. A. Adamos, N. Laskaris, D. Spinellis, and L. Angelis, "Towards an affordable brain computer interface for the assessment of programmers' mental workload," *International Journal of Human-Computer Studies*, vol. 115, pp. 52–66, 2018.

[23] R. Couceiro, R. Barbosa, J. Duráes, G. Duarte, J. Castelhano, C. Duarte, C. Teixeira, N. Laranjeiro, J. Medeiros, P. Carvalho, M. Castelo Branco, and H. Madeira, "Spotting problematic code lines using nonintrusive programmers' biofeedback," in *Proceedings of the 30th International Symposium on Software Reliability Engineering*, ser. ISSRE '19. IEEE, 2019, pp. 93–103.

[24] S. Lee, D. Hooshyar, H. Ji, K. Nam, and H. Lim, "Mining biometric data to predict programmer expertise and task difficulty," *Cluster Computing*, vol. 21, no. 1, pp. 1097–1107, Mar 2018.

[25] J. Peirce, J. R. Gray, S. Simpson, M. MacAskill, R. Höchenberger, H. Sogo, E. Kastman, and J. K. Lindeløv, "Psychopy2: experiments in behavior made easy," *Behavior Research Methods*, vol. 51, no. 1, pp. 195–203, Feb 2019.

[26] J. Siegmund, C. Kästner, S. Apel, A. Brechmann, and G. Saake, "Experience from measuring program comprehension - toward a general framework," in *Software Engineering 2013*, S. Kowalewski and B. Rumpe, Eds. Bonn: Gesellschaft für Informatik e.V., 2013, pp. 239–257.

[27] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): results of empirical and theoretical research," in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, vol. 52, pp. 139–183.