

Investigating Brain Activation Patterns in Human-Human versus Human-AI Pair Programming

Mihaela Y. Hristova¹

¹*IT university of Copenhagen, Rued Langgaards Vej 7, Copenhagen, 2300, Denmark*

Abstract

Programming demands high cognitive effort, involving analytical thinking to understand problems and synthetic thinking to build solutions. Pair programming is a well-established collaborative practice that combines expertise across levels. With AI assistance, engineers take on an additional activity: evaluating, validating, and refining AI-generated code. This new collaboration paradigm raises questions about how programming with an AI assistant differs from working with a human partner at the neural and cognitive level. It also raises a problem for empirical software engineering: cognitive-load measures validated on individual programming may not work the same way in AI-assisted programming.

This PhD addresses that gap by studying brain activation patterns during programming in two contexts: human-human collaboration and AI assistance through a multi-modal approach combining electroencephalography (EEG), electrodermal activity (EDA), eye-tracking, and heart rate variability (HRV). An initial 60-participant crossover experiment provides preliminary evidence: EEG theta/alpha ratio and blink rate differ between AI-assisted and individual programming, whereas EDA and HRV do not.

The resulting data will contribute novel insights into the neural and cognitive dynamics when programming, and into how software engineers engage differently with human versus AI partners, providing neuroscience, human-computer interaction, and software engineering (SE) with practical grounding for designing AI-assisted programming tools and collaborative learning environments.

Keywords

Empirical Software Engineering, Electroencephalography (EEG), Pair Programming, AI-assisted Programming, Cognitive Load, Hyperscanning

1. Introduction

Programming is a complex cognitive task that demands problem-solving, memory, and sustained attention [1]. Existing research has examined the cognitive processes involved in learning to code [2, 3, 4]. Beyond cognition, neuroscience research has identified key brain regions, such as the prefrontal cortex and parietal lobes, that are activated during code comprehension [5] and how neural activation patterns vary according to expertise [6]. The introduction of GitHub Copilot¹ and ChatGPT² has rapidly transformed professional SE, making AI coding assistants a common fixture in professional development environments [7, 8]. The growing role of AI in programming has introduced a new collaboration paradigm, which raises the question of how working with an AI partner differs from working with a human one at the neural and cognitive level. At a behavioral level, the picture remains unclear: a recent systematic review informs that the impact of AI support depends heavily on context, task type, and how developers engage with the tools [9]. When behavioral evidence from AI-assisted programming shows mixed results, understanding what happens at the neural and cognitive level becomes even more relevant.

Brain-imaging techniques such as EEG offer new opportunities to study real-time brain activity during collaborative tasks [10]. However, neuroimaging alone captures only a fraction of the cognitive

IDOSE'26: International Doctoral Symposium on Empirical Software Engineering, October 07, 2026, München, Germany

✉ mihir@itu.dk (M. Y. Hristova)

ORCID  0000-0002-0877-7063 (M. Y. Hristova)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹GitHub, “Introducing GitHub Copilot: Your AI Pair Programmer,” GitHub Blog, June 2021, <https://github.blog/2021-06-29-introducing-github-copilot-ai-pair-programmer/>.

²OpenAI, “ChatGPT: Optimizing Language Models for Dialogue,” OpenAI Blog, November 2022, <https://web.archive.org/web/20221130/https://openai.com/blog/chatgpt/>.

experience. To build a holistic profile of the software engineer, neural metrics must be paired with physiological data. Recent work in SE has demonstrated that combining EEG with EDA, HRV, and eye-tracking produces more robust characterizations of cognitive load during programming than any single signal alone [11, 12, 13]. This research examines both AI-assisted and human–human pair programming compared to individual programming as a baseline.

1.1. Problem statement

This research raises the following research question (RQ): **How do neural and cognitive patterns differ across individual, AI-assisted, and human–human programming?** It is addressed through three sub-questions.

- **RQ1:** How does AI assistance change neural and cognitive patterns within a single developer, relative to individual programming?
- **RQ2:** Do cognitive-load measures validated on individual programming remain valid under AI assistance?
- **RQ3:** How do neural and cognitive patterns during human–human pair programming compare with those under AI assistance?

These RQs will be answered by combining EEG, EDA, HRV, and eye-tracking to capture cognitive load, arousal, and attention. Individual programming, in which a developer works alone, serves as the baseline; AI-assisted programming, in which a developer works with an AI coding assistant, is the first collaboration mode; and human–human pair programming, in which two developers collaborate, is the second. The research investigates two levels of analysis: cognitive load within a single brain, and synchrony between two brains during pair programming.

1.2. Background

EEG is a non-invasive neuroimaging technique that records electrical potentials generated by synchronized postsynaptic activity of cortical neurons, measured via electrodes placed on the scalp [12]. Its strength is a millisecond-scale temporal resolution, which allows researchers to track rapid neural dynamics that slower techniques cannot resolve [13]. In SE research, EEG has been used to identify cognitive load during code comprehension, with theta and alpha band power emerging as the most reliable markers of mental effort [11, 12].

Hyperscanning refers to the simultaneous measurement of brain activity across multiple individuals during social interaction [14]. It has been applied to study neural synchrony in decision-making, education, and social interaction [15, 16, 17], revealing that inter-brain synchrony is a reliable marker of successful collaboration. Hyperscanning can be implemented with fMRI, EEG or fNIRS [15]. EEG-based hyperscanning is particularly well suited to programming contexts given its temporal resolution and compatibility with naturalistic task settings [10, 18].

Biometric signals. Beyond EEG, peripheral physiological signals serve as reliable proxies for cognitive load and arousal. *EDA* reflects sympathetic nervous system activation and has been used in SE research to differentiate between stress and cognitive load during code comprehension [11, 19]. Similarly, *HRV* captures autonomic balance, decreasing as cognitive demand increases to mark mental effort [20, 21]. *Eye-tracking* complements these signals by capturing attentional processes through fixation duration and pupil dilation, which map directly to developer expertise [22, 23]. No single signal provides a complete picture: each captures a different part of the cognitive response, and together they estimate mental state more reliably than any single channel alone [12, 13].

Collaborative programming. Collaboration in this research takes two forms: AI-assisted programming and human–human pair programming. Pair programming can enhance code quality and efficiency, though the benefit depends on task complexity and programmer expertise [24], and it can cost roughly 15% more time than individual work [25]. These limitations motivate a deeper investigation into how the

two collaboration modes differ at the neural and cognitive level, a difference that behavioral measures alone cannot fully capture.

AI-assistance in programming. AI coding assistants are software tools that leverage large language models to support developers, offering capabilities such as code completion, generation, and explanation. They are now widely adopted in professional settings and their use continues to grow [26, 27]. Despite this, their cognitive and neural impact on developers remains poorly understood.

2. Related literature

AI-assisted programming. AI copilots are reshaping knowledge transfer practices within teams, replacing traditional pair programming dynamics with human-AI collaboration patterns [28]. AI coding assistants are also defined as collaborative teammates rather than passive tools [29, 30]. Early insights into AI-assisted pair programming indicate a shift in developer roles from code writing to reviewing AI-generated suggestions [31]. Verification of AI-generated suggestions demands significant cognitive effort [32], and growing concerns about dependency and skill atrophy [33, 34] further motivate the need to understand what is happening in the brain during AI-assisted programming. Ma et al. [35] compare human-human and human-AI pair programming at the behavioral level, and Jang et al. [36] analyze communication patterns across both conditions. However neither examines the neural mechanisms underlying these differences.

Neural and physiological measurement in SE. Peitek et al. [4] used fMRI to identify brain regions activated during code comprehension, while Ikutani et al. [6] showed that expert programmers have distinct cortical representations of source code compared to novices. Beyond neuroimaging, biometric signals have been used to classify developer task types [19], predict bugs from behavioral and physiological data [37], and estimate developer interruptibility in field settings [38]. EEG has been applied directly to programming activity to compare developers' brainwave activity across solo, pair, and mob programming sessions, using attention and meditation indices to distinguish the cognitive profiles of the three modes [39]. However, all of these studies examine human programming configurations, none has investigated the neural dynamics of AI-assisted pair programming, which is the focus of Phase 1.

Hyperscanning was first studied in a 1965 EEG study of twins showing correlated alpha activity [40]. It has since been applied to cooperation, consistently finding elevated interpersonal neural synchronization during joint work in prefrontal and right temporoparietal regions [41, 42], and closer to education, in dyads solving collaborative problems [43]. An EEG study of pair programming linked coordination episodes to systematic EEG changes, particularly in frontal theta dynamics and event-related responses, though the signal was recorded from only one member of the pair [44]. A recent fNIRS hyperscanning study of pair programming found that pairing by prior knowledge shaped synchrony in the same prefrontal and temporoparietal regions reported in earlier cooperation studies [45]. Beyond these studies, the neural basis of pair programming remains almost entirely unexplored, particularly with EEG and in human-AI settings, which is the focus of Phase 2.

3. Research approach

The research conducted throughout this PhD is organized in two phases. Each phase contains specific actions with theoretical and empirical contributions. Both phases work toward a single goal: understanding how AI-assisted and human-human collaboration differ and overlap at the neural level.

Phase 1: Investigating neural correlates in AI-assisted programming. This phase investigates the neural correlates of individual and AI-assisted programming, combining inductive interviews with a deductive multi-modal crossover experiment (EEG, EDA, HRV, eye-tracking) in which each participant solves one task with an AI assistant (ChatGPT) and one without.

Phase 2: Investigating neural correlates in human-human programming. This phase compares AI-assisted with human-human pair programming by applying EEG-hyperscanning to measure

inter-brain synchrony during collaborative programming, alongside behavioral metrics such as task completion time, code quality, and participant feedback.

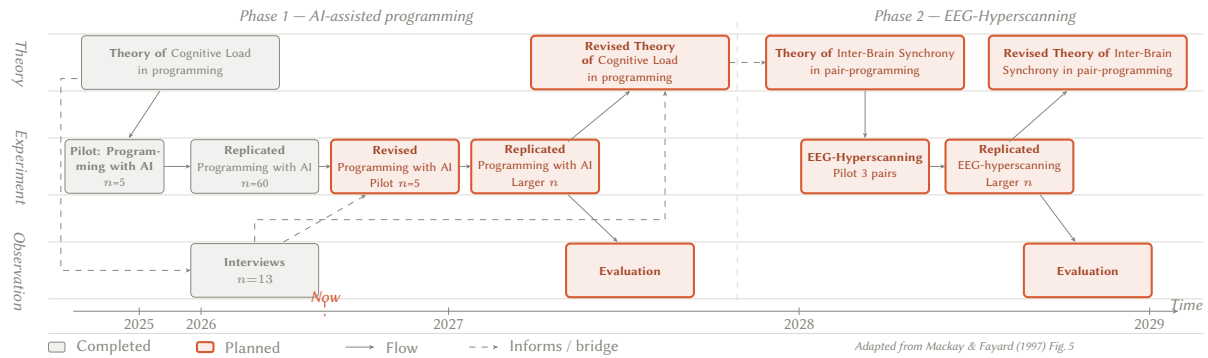


Figure 1: PhD research framework: from AI-assisted to collaborative programming

Figure 1 maps these two phases onto Mackay and Fayard’s [46] triangulation model with Phase 1 on the left and Phase 2 on the right. The research design integrates two methodological traditions: inductive and deductive. The inductive path in Phase 1 begins with semi-structured interviews with 13 professional software developers. Interviews inform how AI assistants are used in real workflows: which tasks are delegated to AI, how suggestions are evaluated and verified, when AI is accepted versus rejected, and the perceived effort and trust involved. The interviews are the Observation component of the model, and they feed the other two: the Experiment, through the design of realistic tasks and conditions, and the Theory, through a first revision of the cognitive-load model for AI-assisted programming. Because the tasks come from observed practice, the lab-based crossover gains ecological validity. However, laboratory setting cannot fully reproduce everyday development: time pressure, tooling preferences, and self-directed workflows are constrained, so the results approximate rather than replicate real AI-assisted practice.

The deductive path comprises a pilot and a subsequent within-subjects crossover experiment with 60 participants, measuring physiological signals under AI-assisted and individual programming conditions to test predefined hypotheses. A distinguishing feature of the study design is the pilot preceding a full replicated experiment. Participant feedback, combined with a revisited literature review, informs a refined design before the main study is conducted. As earlier work shows that the effect of AI assistance depends on both the programming task and the surrounding context, the design controls each of these rather than leaving them to vary. The two task scenarios are matched for difficulty and randomized in order across participants, so any signal difference reflects the AI assistant, not the task. Contextual factors are addressed on two levels: the within-subjects crossover design controls for individual differences and order effects, since each participant completes both conditions and serves as their own baseline; and the experimental environment (hardware, IDE, and session protocol) is standardized across conditions so that setting does not covary with the manipulation.

Inter-brain synchrony theory motivates Phase 2 because pair programming is a sustained joint activity, and hyperscanning studies have shown that people who cooperate effectively tend to exhibit synchronized brain activity. Whether this extends to collaborative programming, and how it relates to what happens during AI-assisted programming, is largely unexplored. Phase 2 uses EEG-hyperscanning to examine whether human–human pair programming and AI-assisted programming share common neural signatures, or whether they are fundamentally distinct. A systematic literature review will guide the design of the Phase 2 pilot, which will precede a full experiment.

The phases are kept separate because they ask different measurement questions. Phase 1 concerns a single brain and its cognitive load, whereas Phase 2 concerns two brains and the synchrony between them, which requires EEG-hyperscanning and a different analysis. Running both as one experiment would make it hard to tell whether the result reflects the single-brain or inter-brain question. Each phase ends with an evaluation step. In Phase 1, this means running statistical tests on the full multimodal

dataset and checking whether the cognitive-load measures still hold under AI assistance, which leads to a revised cognitive-load model. In Phase 2, the inter-brain synchrony results are interpreted alongside behavioral metrics such as task completion time, code quality, and participant feedback, so the neural findings are not read on their own. These evaluation steps are where each phase feeds back into the theory it started from, closing the triangulation loop shown in Figure 1.

4. Current progress and future directions

The research programme is currently mid-way through Phase 1. The completed pilot (n=5) and the subsequent replicated within-subjects crossover experiment (n=60) covered two conditions: AI-assisted and individual programming (Figure 2), with two scenarios both similar in difficulty. Task instructions were given in UML and activity diagrams to motivate participants to write their own instructions (copy-paste of pictures was not allowed). Task duration was capped at 20 minutes and the complexity required significant cognitive load to be completed. Each of the two task scenarios (implementing the connection logic of a mobile device, and building a simplified online shopping application) was paired with one of the two conditions, and the order of the two tasks was randomized across participants. The experiment was executed in collaboration with the University of Bari, with participants drawn in equal numbers from different degree programmes. It tested four pre-registered hypotheses covering AI's effect on physiological measures, the role of developer experience, the link to performance, and the match between perceived and actual cognitive state, analyzed with linear mixed-effects models to account for repeated measurements per participant. Findings show that EEG theta/alpha ratio and blink rate differed significantly between AI-assisted and individual programming conditions, while EDA and HRV did not, calling for re-validation of existing cognitive-load measures in AI-assisted contexts [47]. Phase 1 still requires a revised experiment informed by the interviews, a larger-sample replication, and the resulting revised cognitive-load model.

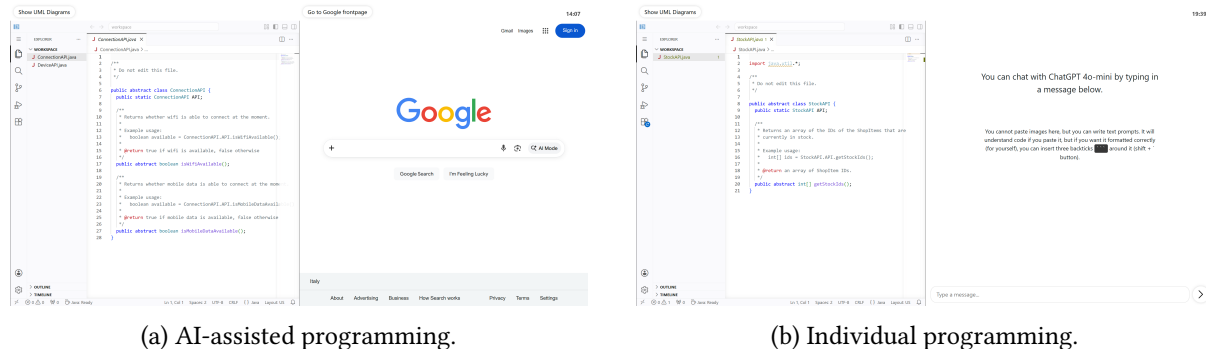


Figure 2: Development environment for experimental conditions

For Phase 2, the revised cognitive-load model from Phase 1, together with a systematic literature review, will formulate hypotheses on inter-brain synchrony across the three conditions — for instance, whether the conditions that shift single-brain load in Phase 1 also correspond to differences in inter-brain synchrony. A dedicated EEG-hyperscanning pilot will precede a full replication experiment with a larger sample, enabling comparison of inter-brain synchrony across all three conditions.

The primary contribution of the research is empirical: one of the first neurophysiological comparisons of AI-assisted, individual, and human–human programming, using multimodal biometrics and EEG-hyperscanning. A secondary methodological and theoretical contribution follows from it: the findings indicate that cognitive-load measures validated on individual programming may not transfer to AI-assisted contexts, motivating a revised model. Practically, knowing when AI assistance lowers or raises cognitive load indicates where in the workflow AI is best introduced e.g. timing suggestions so developers retain enough cognitive capacity to validate AI-generated code rather than accept it uncritically.

Declaration on Generative AI

Declaration of AI use: During the preparation of this work, the author used Claude, in order to: improve readability and grammar, spelling check, paraphrase and reword where needed, as well as formatting of Figure 1. After using Claude, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] E. Soloway, K. Ehrlich, Empirical studies of programming knowledge, *IEEE Transactions on Software Engineering* SE-10 (1984) 595–609. doi:10.1109/TSE.1984.5010283.
- [2] J. Waite, S. Sentance, Teaching Programming in Schools: A Review of Approaches and Strategies, Technical Report, Raspberry Pi Foundation, 2021. URL: <https://www.raspberrypi.org/app/uploads/2021/11/Teaching-programming-in-schools-pedagogy-review-Raspberry-Pi-Foundation.pdf>.
- [3] M. Endres, Z. Karas, X. Hu, I. Kovelman, W. Weimer, Relating reading, visualization, and coding for new programmers: A neuroimaging study, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), IEEE, 2021, pp. 600–612. doi:10.1109/ICSE43902.2021.00062.
- [4] N. Peitek, J. Siegmund, S. Apel, C. Kästner, C. Parnin, A. Bethmann, T. Leich, G. Saake, A. Brechmann, A Look into Programmers' Heads, *IEEE Transactions on Software Engineering* 46 (2020) 442–462. doi:10.1109/TSE.2018.2863303.
- [5] A. A. Ivanova, S. Srikant, Y. Sueoka, H. H. Kean, R. Dhamala, U.-M. O'Reilly, M. U. Bers, E. Fedorenko, Comprehension of computer code relies primarily on domain-general executive brain regions, *eLife* 9 (2020) e58906. doi:10.7554/eLife.58906.
- [6] Y. Ikutani, T. Kubo, S. Nishida, H. Hata, K. Matsumoto, K. Ikeda, S. Nishimoto, Expert programmers have fine-tuned cortical representations of source code, *eNeuro* 7 (2020) ENEURO.0405–20.2020. doi:10.1523/ENEURO.0405–20.2020.
- [7] M. Merler, R. Pan, R. Krishna, T. K. Ho, R. Pavuluri, M. Vuković, Usage, effects and requirements for ai coding assistants in the enterprise: An empirical study, in: Proceedings of the 3rd International Workshop on Large Language Models for Code (LLM4Code '26), ACM, New York, NY, USA, 2026, pp. 97–104. URL: <https://doi.org/10.1145/3786181.3788727>. doi:10.1145/3786181.3788727.
- [8] J. T. Liang, C. Yang, B. A. Myers, A large-scale survey on the usability of ai programming assistants: Successes and challenges, in: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), ACM, Lisbon, Portugal, 2024, pp. 616–628. doi:10.1145/3597503.3608128.
- [9] A. Mohamed, M. Assi, M. Guizani, The impact of llm-assistants on software developer productivity: A systematic literature review, 2025. doi:10.48550/arXiv.2507.03156.
- [10] A. Zamm, J. D. Loehr, C. Vesper, et al., A practical guide to EEG hyperscanning in joint action research: from motivation to implementation, *Social Cognitive and Affective Neuroscience* 19 (2024) nsae015. doi:10.1093/scan/nsae026.
- [11] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, M. Züger, Using psycho-physiological measures to assess task difficulty in software development, in: Proceedings of the 36th International Conference on Software Engineering (ICSE 2014), ACM, 2014, pp. 402–413. doi:10.1145/2568225.2568266.
- [12] J. Medeiros, M. Simões, J. Castelhana, R. Abreu, R. Couceiro, J. Henriques, M. Castelo-Branco, H. Madeira, C. Teixeira, P. de Carvalho, EEG as a potential ground truth for the assessment of cognitive state in software development activities: A multimodal imaging study, *PLOS ONE* 19 (2024) e0299108. doi:10.1371/journal.pone.0299108.
- [13] N. Peitek, J. Siegmund, S. Apel, Correlates of programmer efficacy and their link to experience: A combined EEG and eye-tracking study, in: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), ACM, 2022, pp. 1637–1648. doi:10.1145/3540250.3549084.
- [14] P. R. Montague, G. S. Berns, J. D. Cohen, S. M. McClure, G. Pagnoni, M. Dhamala, M. C. Wiest,

- I. Karpov, R. D. King, N. Apple, R. E. Fisher, Hyperscanning: simultaneous fmri during linked social interactions, *Neuroimage* 16 (2002) 1159–1164. doi:10.1006/nimg.2002.1150.
- [15] M. Y. Wang, P. Luan, J. Zhang, Y. T. Xiang, H. Niu, Z. Yuan, Concurrent mapping of brain activation from multiple subjects during social interaction by hyperscanning: a mini-review, *Quantitative Imaging in Medicine and Surgery* 8 (2018) 819–837. doi:10.21037/qims.2018.09.07.
- [16] S. R. Dumas G., Nadel J., Inter-brain synchronization during social interaction, *PLoS ONE* 5 (2010) e12166. doi:10.1371/journal.pone.0012166.
- [17] W.-P. T. S H Jessica Tan, Jin Nen Wong, Is neuroimaging ready for the classroom? a systematic review of hyperscanning studies in learning, *NeuroImage* 280 (2023) 120325. doi:10.1016/j.neuroimage.2023.120367.
- [18] A. Czeszumski, S. Eustergerling, A. Lang, et al., Hyperscanning: a valid method to study neural inter-brain underpinnings of social interaction, *Frontiers in Human Neuroscience* 14 (2020) 39. doi:10.3389/fnhum.2020.00039.
- [19] D. Fucci, D. Girardi, N. Novielli, L. Quaranta, F. Lanubile, A replication study on code comprehension and expertise using lightweight biometric sensors, in: *Proceedings of the 27th International Conference on Program Comprehension (ICPC 2019)*, IEEE, 2019, pp. 311–322. doi:10.1109/ICPC.2019.00050.
- [20] J. Aasman, G. Mulder, L. J. M. Mulder, Operator effort and the measurement of heart-rate variability, *Human Factors* 29 (1987) 161–170. doi:10.1177/001872088702900204.
- [21] B. Cinaz, B. Arnrich, R. La Marca, G. Tröster, Monitoring of mental workload levels during an everyday life office-work scenario, *Personal and Ubiquitous Computing* 17 (2013) 229–239. doi:10.1007/s00779-011-0466-1.
- [22] S. D. Aljehane, B. Sharif, J. I. Maletic, Studying developer eye movements to measure cognitive workload and visual effort for expertise assessment, *Proceedings of the ACM on Human-Computer Interaction* 7 (2023) 1–18. doi:10.1145/3591135.
- [23] T. Alakmeh, D. R. Reich, L. A. Jäger, T. Fritz, Predicting code comprehension: A novel approach to align human gaze with code using deep neural networks, *Proceedings of the ACM on Software Engineering* 1 (2024) 1982–2004. doi:10.1145/3660795.
- [24] E. Arisholm, H. Gallis, T. Dybå, D. I. K. Sjøberg, Evaluating pair programming with respect to system complexity and programmer expertise, *IEEE Transactions on Software Engineering* 33 (2007) 65–86. doi:10.1109/TSE.2007.17.
- [25] J. E. Hannay, T. Dybå, E. Arisholm, D. I. K. Sjøberg, The effectiveness of pair programming: A meta-analysis, *Information and Software Technology* 51 (2009) 1110–1122. doi:10.1016/j.infsof.2009.02.001.
- [26] Google Cloud DORA Team, D. Smith, M. Villamejor, A. Harrison, 2025 State of AI-Assisted Software Development Report: AI as an Amplifier, Research Report, Google Cloud Research, 2025. URL: <https://dora.dev/dora-report-2025/>.
- [27] Microsoft and LinkedIn, 2025 Work Trend Index Annual Report: The Frontier Firm Is Born, Technical Report, Microsoft Corporation, 2025. URL: <https://www.microsoft.com/en-us/worklab/work-trend-index/2025-the-year-the-frontier-firm-is-born>.
- [28] A. Welter, N. Schneider, T. Dick, K. Weis, C. Tinnes, M. Wyrich, S. Apel, From developer pairs to ai copilots: A comparative study on knowledge transfer, 2025. Preprint.
- [29] H. Li, H. Zhang, A. E. Hassan, The rise of ai teammates in software engineering (se) 3.0: How autonomous coding agents are reshaping software engineering, *arXiv preprint arXiv:2507.15003* (2025). arXiv:2507.15003.
- [30] S. Stieber, From code production to code orchestration: How ai coding assistants are reshaping software engineering, 2026. URL: <https://doi.org/10.13140/RG.2.2.13510.95048>. doi:10.13140/RG.2.2.13510.95048, preprint, ResearchGate, CC BY 4.0.
- [31] C. Bird, D. Ford, T. Zimmermann, N. Forsgren, E. Kalliamvakou, T. Lowdermilk, I. Gazit, Taking flight with copilot: Early insights and opportunities of ai-powered pair-programming tools, *Queue* 20 (2023) 35–57. URL: <https://doi.org/10.1145/3582083>. doi:10.1145/3582083.
- [32] H. Mozannar, G. Bansal, A. Fourney, E. Horvitz, Reading between the lines: Modeling user behavior

- and costs in ai-assisted programming, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24), ACM, 2024, pp. 1–16. doi:10.1145/3613904.3641936.
- [33] J. Prather, “it’s weird that it knows what i want”: Usability and interactions with copilot for novice programmers, *ACM Transactions on Computer-Human Interaction* 31 (2024) 1–31. doi:10.1145/3617367.
- [34] H.-P. Lee, A. Sarkar, L. Tankelevitch, I. Drosos, S. Rintel, R. Banks, N. Wilson, The impact of generative ai on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers, in: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, CHI '25, ACM, New York, NY, USA, 2025. doi:10.1145/3706598.3713778.
- [35] Q. Ma, T. Wu, K. Koedinger, Is AI the better programming partner? Human-Human pair programming vs. Human-AI pAIr programming, 2023. doi:10.48550/arXiv.2306.05153. arXiv:2306.05153.
- [36] W. Jang, R. Hou, H. Gao, E. Kasneci, Analyzing communication logs in pair programming: A comparison of human- and llm-based approaches, 2024. doi:10.1109/ITHET61869.2024.10837662.
- [37] G. Laudato, S. Scalabrino, N. Novielli, F. Lanubile, R. Oliveto, Predicting bugs by monitoring developers during task execution, in: Proceedings of the 45th International Conference on Software Engineering (ICSE '23), IEEE, 2023, pp. 1110–1122. doi:10.1109/ICSE48619.2023.00100.
- [38] M. Züger, T. Fritz, Interruptibility of software developers and its prediction using psychophysiological sensors, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI 2015), ACM, 2015, pp. 2981–2990. doi:10.1145/2702123.2702593.
- [39] M. Shiraishi, H. Washizaki, D. Saito, Y. Fukazawa, Comparing participants’ brainwaves during solo, pair, and mob programming, in: P. Gregory, C. Lassenius, X. Wang, P. Kruchten (Eds.), *Agile Processes in Software Engineering and Extreme Programming (XP 2021)*, volume 419 of *Lecture Notes in Business Information Processing*, Springer, 2021. doi:10.1007/978-3-030-78098-2_13.
- [40] T. D. Duane, T. Behrendt, Extrasensory electroencephalographic induction between identical twins, *Science* 150 (1965) 367. doi:10.1126/science.150.3694.367.
- [41] X. Cui, D. M. Bryant, A. L. Reiss, NIRS-based hyperscanning reveals increased interpersonal coherence in superior frontal cortex during cooperation, *NeuroImage* 59 (2012) 2430–2437. doi:10.1016/j.neuroimage.2011.09.003.
- [42] R. Li, N. Mayseless, S. Balters, A. L. Reiss, Dynamic inter-brain synchrony in real-life inter-personal cooperation: A functional near-infrared spectroscopy hyperscanning study, *NeuroImage* 238 (2021) 118263. doi:10.1016/j.neuroimage.2021.118263.
- [43] P. D. Antonenko, R. Davis, J. Wang, M. Celepkolu, On the same wavelength: Exploring team neurosynchrony in undergraduate dyads solving a cyberlearning problem with collaborative scripts, *Mind, Brain, and Education* 13 (2019) 4–13. doi:10.1111/mbe.12187.
- [44] N. Tulkunova, A. Bogapova, M. Snigireva, A. Tormasov, E. Zuev, Toward EEG-based characterization of cognitive coordination in collaborative software implementation: A pilot study, in: The 9th International Conference on Information and Computer Technologies (ICICT 2026), ACM, Honolulu, HI, USA, 2026, pp. 1–5. URL: <https://doi.org/10.1145/3803291.3803327>. doi:10.1145/3803291.3803327.
- [45] Q. Yang, Z. Zhan, H. Mei, S. Cai, Z. Dong, Z. Lin, X. Zhang, L. Luo, B. Xu, W. Li, Interpersonal neural synchronization in pair programming based on prior knowledge: a fNIRS hyperscanning study, *International Journal of Computer-Supported Collaborative Learning* (2026). URL: <https://doi.org/10.1007/s11412-026-09480-z>. doi:10.1007/s11412-026-09480-z, advance online publication.
- [46] W. E. Mackay, A.-L. Fayard, HCI, natural science and design: A framework for triangulation across disciplines, in: Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '97), ACM, 1997. doi:10.1145/263552.263612.
- [47] P. Burelli, F. Calefato, D. Grassi, M. Y. Hristova, N. Novielli, A. A. Romano, P. Tell, Using biometrics to understand AI-assisted coding performance and its perception, *Empirical Software Engineering* (2025). Manuscript under review.