Creative Activity Traces in Coding: A Proposed Methodology to Study Creativity of Novice Programmers Using GenAI

VITTORIA FRAU, Aarhus University, Denmark

SAMANGI WADINAMBIARACHCHI, School of Computing and information system, University of Melbourne, Australia

JONAS FRICH, Aarhus University, Denmark

CCS Concepts: • **Human-centered computing → HCI theory, concepts and models**.

Additional Key Words and Phrases: Creativity, GenAI, Programmers

ACM Reference Format:

Vittoria Frau, Samangi Wadinambiarachchi, and Jonas Frich. 2026. Creative Activity Traces in Coding: A Proposed Methodology to Study Creativity of Novice Programmers Using GenAI. In *Proceedings of CHI*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction and Context

Approximately five months ago, one of the authors attended a Dagstuhl Seminar[1] on *Creativity, GenAI, and Software Development* [11] based on prior research on creativity support tools in HCI. The week brought together promising study ideas, ambitious proposals, and recurring, unresolved questions. One question in particular continued to resonate after the seminar concluded: *How should creativity in software development be defined and analyzed?*

At first glance, existing frameworks appear to provide robust answers. The Consensual Assessment Technique (CAT) [1][2] addresses evaluation of the *product* [13]: an artifact judged to be novel, appropriate, useful or valuable by a suitably knowledgeable social group [7, 15]. Linkography and related process analytics focus on the *process* [6, 12]. Instruments such as the Creativity Support Index (CSI) [5] examine the *press*, i.e. the tools and environment that shape creative work. Together, these approaches represent solid, time-tested ideas of creativity.

However, upon reflection, there was a question of whether these approaches fully capture the essence of the creative process involved in coding, especially for learners. The term "coding" is used here deliberately to reflect the activity of the cohort under study: novice learners attempting to make some code work, rather than professional software engineers operating within established practices.

[1]Dagstuhl Seminars are invitation-only, week-long research meetings at Schloss Dagstuhl (Leibniz Center for Informatics, Germany) where groups take stock of a topic and try to set a research agenda together.
[2]Which, ironically, is also abbreviated "CAT."

Authors' Contact Information: Vittoria Frau, vittoria.frau@cc.au.dk, Aarhus University, Aarhus, Denmark; Samangi Wadinambiarachchi, samangi.w@unimelb.edu.au, School of Computing and information system, University of Melbourne, Melbourne, Australia; Jonas Frich, frich@cc.au.dk, Aarhus University, Aarhus, Denmark.

Manuscript submitted to ACM

Consider a beginner who struggles getting comfortable with JavaScript's `Map`. Instead of adopting the canonical structure, they create an array of objects with two properties (`name` and `value`) and implement a simple loop to enforce uniqueness. The resulting code may not be "novel" in a domain-level sense, yet it *is* locally creative and entirely appropriate. It's a workaround that honors constraints.

The distinction between creativity in the *process* and judged creativity in the *product* is well established [13]. When examining non-professional coders, the discussion also intersects with notions of small-c and Pro-c creativity [8]. Yet these perspectives may miss a middle layer: the fine-grained, in-editor traces where creativity manifests for novices, and where GenAI systems may either support or constrain it.

This paper proposes a methodology to capture and analyze those *creative activity traces in coding*: edits, runs, refactors, branches, prompts, acceptances, and the tiny transformations that turn "it almost works" into "it works." The agenda is twofold. First, we advance a methodological contribution by identifying measures that can reliably capture creativity in-action among novice programmers using GenAI tools. Second, we focus on learning: exercising creativity feels good, sustains momentum, and often helps to get mitigate some of the frustration that especially beginners face. By making these dynamics visible, we aim to inform the design of better creativity support tools and more supportive learning experiences.

2 The Proposed Study

The study that we have initiated investigates how AI-assisted programming reshapes creative activity in undergraduate coding contexts. Rather than evaluating creativity solely through final artifacts, we focus on creative activity traces: the fine-grained edits, iterations, restructurings, prompts, and transformations that occur during coding.

2.1 Context and Participants

The study will be conducted in the context of two courses (Backend and Aesthetic Programming) that both use JavaScript as their primary programming language, providing a shared technical foundation for comparative analysis. Approximately 80 students in total are involved, with around 40 students enrolled in a Backend Course, and around 40 students enrolled in an Aesthetic Programming course.

Although both courses involve programming in JavaScript, they emphasize very different forms of coding: In the Backend course, students primarily engage in what we can be considered studies of the creative process on a micro level [4]: the focus is on how they employ creative thinking to structure code, implement functionality, and work around gaps in knowledge and skills (which is sure to be there in novice coders).

In contrast, the Aesthetic Programming course emphasizes what we consider product-level creativity [13], where the primary focus lies on the aesthetic and experiential qualities of the output that is produced. Assignments often involve visual or interactive sketches (e.g., using p5.js), where creativity is evaluated in terms of expressive output rather than architectural structure. This distinction allows us to explore how creative activity traces manifest differently depending on whether the emphasis is on process-level problem solving or artifact-level expression.

2.2 Study Design

The study consists of two sequential stages, which will triangulate between the activities of observation, theory-building, prototyping before returning to observations [10]. First, a formative study (i) will investigate how these particular students currently use AI tools in their programming tasks. Informed by these findings and prior work on interaction design for programming environments [9] we will develop a small prototype that modifies how AI is integrated into

their coding practice. We then use the prototype to conduct a comparative study (ii) across both different configurations of AI support and in two different coding contexts.

The prototype will experiment with alternative interaction patterns around AI assistance (e.g., providing multiple alternative solutions instead of a single answer, inserting structured gaps in generated code to encourage completion, or prompting incremental refinement instead of full-solution generation). The goal is to examine how subtle changes in AI integration reshape students' creative processes and decision-making in the editor.

2.2.1 Formative Study. We will recruit five students from each course (N = 10). Each session will last approximately 30 minutes and follow a semi-structured interview protocol combined with a short programming task using a think-aloud procedure. The purpose of this phase is to understand how students currently integrate AI tools into their programming workflows, both individually and collaboratively, and use this as inspiration for building our conceptual model of what the prototype will do.

2.2.2 Comparative Study. We will conduct a comparative study involving approximately 10 groups of students (across both courses). Each group will participate in one 1-hour sessions.

The study adopts a within-subject design. Within the 1-hour session, each group will work under two conditions: a baseline condition and an experimental condition, each lasting approximately 30 minutes. In the baseline condition, students will use AI tools according to their current practices, as identified in the formative study (e.g., web-based chat interfaces or IDE-integrated assistants). In the experimental condition, students will work with our prototype implementing alternative interaction patterns around AI assistance. To mitigate order effects and learning transfer between conditions, we will randomize order. The intention is to measure the product (i.e. what they produced with code), and the process or creative activity traces of writing the code in both of the courses.

2.3 Proposed Metrics

To measure the creativity of the product we propose some version of the previously mentioned consensual assessment technique [1]. One additional avenue might be automated assessments, where it has recently been demonstrated that modern LLMs, when guided by a comparative framework, seems effective for some creative domains (poetry) [14].

In order to analyze participant process we plan on assessing the code it self and to treat these as creative activity traces following the methodology recently proposed by Amini et al. [2]. Here the idea is to use Boden's [3] creativity framework to classify each approach at solving a task as either exploratory (working within the provided code examples), combinatorial (integrating additional aesthetic or interactive elements), or transformational (substantially rewriting or extending the code). We hope that coupling this with think-aloud protocols will allow us to characterize the different creative processes and strategies participants employed, and how they were impacted by the features of the AI and the prototype.

3 Participation in the Workshop

If this position paper is accepted, the last author will attend the workshop. We hope that our proposed study can serve as a concrete example within the planned activity of sketching a "technique design space," and we look forward to engaging in fruitful discussions with other participants.

References

[1] Teresa M. Amabile. 1996. *Creativity in Context.* Westview Press.

[2] Mahta Amini, Jay Olson, and Zohreh Sharafi. 2024. Coding with a Creative Twist: Investigating the Link Between Creativity Scores and problem-solving Strategies. In *Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering: New Ideas and Emerging Results*. 21–25.

[3] Margaret A Boden. 2010. *Creativity and art: Three roads to surprise.* Oxford University Press.

[4] Marion Botella, Julien Nelson, and Franck Zenasni. 2019. It is time to observe the creative process: How to use a creative process report diary (CRD). *The Journal of Creative Behavior* 53, 2 (2019), 211–221.

[5] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 1–25.

[6] Gabriela Goldschmidt. 2014. *Linkography: Unfolding the Design Process.* MIT Press.

[7] B A Hennessey, T M Amabile, and J S Mueller. 2011. Consensual Assessment. In *Encyclopedia of Creativity (Second Edition)* (second edition ed.), Mark A Runco and Steven R Pritzker (Eds.). Academic Press, San Diego, 253–260. https://doi.org/10.1016/B978-0-12-375038-9.00046-7

[8] James C Kaufman and Ronald A Beghetto. 2009. Beyond big and little: The four c model of creativity. *Review of general psychology* 13, 1 (2009), 1–12.

[9] Majeed Kazemitabaar, Oliver Huang, Sangho Suh, Austin Z Henley, and Tovi Grossman. 2025. Exploring the design space of cognitive engagement techniques with ai-generated code for enhanced learning. In *Proceedings of the 30th international conference on intelligent user interfaces*. 695–714.

[10] Wendy E Mackay and Anne-Laure Fayard. 1997. 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*. 223–234.

[11] Rafael Prikladnicki, Daniel Russo, Margaret-Anne Storey, and André van der Hoek. 2025. Dagstuhl Perspectives Workshop 25412: Creativity, GenAI, and Software Development: A Future Together. https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/25412. Dagstuhl Perspectives Workshop, Oct 05–Oct 10, 2025.

[12] Eric Rawn, Jingyi Li, Eric Paulos, and Sarah E Chasins. 2023. Understanding Version Control as Material Interaction with Quickpose. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[13] Mel Rhodes. 1961. An analysis of creativity. *The Phi delta kappan* 42, 7 (1961), 305–310.

[14] Piotr Sawicki, Marek Grześ, Dan Brown, and Fabrício Góes. 2025. Can large language models outperform non-experts in poetry evaluation? a comparative study using the consensual assessment technique. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 31889–31906.

[15] R Keith Sawyer. 2011. *Explaining creativity: The science of human innovation.* Oxford university press.