"I'm Not Reading All of That": Understanding Software Engineers' Level of Cognitive Engagement with Agentic Coding Assistants

Carlos Rafael Catalan, Samsung R&D Institute Philippines *Lheane Marie Dizon, Samsung R&D Institute Philippines*
Patricia Nicole Monderin, Samsung R&D Institute Philippines *Emily Kuang, York University*

Main theme(s): *assessing and measuring outcomes / adoption / design strategy*

Target domain(s): *programming, critical thinking, decision making*

Cognitive 'target(s)': *critical thinking, user engagement*

Type of contribution & main idea

A formative study examining software engineers' level of cognitive engagement when performing tasks with agentic coding assistants. We aim to answer the following research questions (RQ).

RQ1: How cognitively engaged with the task are software engineers when working with agentic coding assistants?

RQ2: How did they recall, understand, analyze, and evaluate different aspects of their interaction with an ACA?

With this understanding, we propose design considerations for these tools to help users **sustain their engagement**, and **encourage critical thinking** throughout the task.

The AI tool's key characteristics

We use Cline, an open source ACA. Its main functionalities are the following:

Plan Mode:
This is the mode set by default, it analyzes all the tools available to it, the constraints, the user requirements, etc.

Act Mode:
This executes the plan. In a code generation task, the agent writes the code, runs it, and evaluates it according to the requirements. With the Model Context Protocol, the agent can invoke any tools available in the integrated development environment such as the terminal, or the file directory to help with the task

At any point during either of the modes, Cline may ask clarifying questions to the user. **The interaction between the user and Cline is conversational**, where both user and agent communicates through natural language.

Fig 1. Workflow of Cline for a Code Generation Task

How do we define "Cognitive Engagement?"

Bloom's Taxonomy [1] conceptualizes cognition into various levels of complexity and depth. In this conceptual hierarchy, basic information processing such as **Recall/Remembering** and **Understanding** serve as the foundation that enables the individual to perform complex reasoning tasks such as **Applying, Analyzing, Evaluating,** and **Creating.**

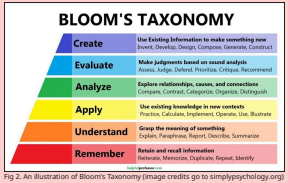


Fig 2. An illustration of Bloom's Taxonomy (image credits go to simplypsychology.org)

For this experiment we only consider the levels of **Recall**, **Understanding**, **Analyzing**, and **Evaluating**. Apply and Create will not be included they require insights from multiple interactions. For our formative study, we will only be considering one code generation scenario for the user.

We use these levels in our self-report survey to understand how cognitively engaged software engineers were to the code generation task while using Cline

How do we measure Cognitive Engagement?

In psychology literature, **self-reporting surveys [2]** serve as a reliable measure of cognitive engagement. These surveys have been used extensively in the education domain to measure students motivations and learning goals. For our use case, we included questions in our survey that would measure how well software engineers Recalled, Understood, Analyzed, and Evaluated various aspects of their interaction with Cline.

Method

We recruited 4 participants from our study. Each participant (P) represents a different category of years of professional software engineering experience: P1: < 1 year, P2: 1-5 years, P3: 6-10 years, P4: > 10 years. Each participant was given a prompt* to use for Cline in a code generation task. The task begins with the user prompting Cline which is set to Plan mode initially. Once the participant is satisfied with the plan, Act mode is initiated to generate the code. The task ends when "Start New Task" appears. The participant is then given a survey to answer, but they are not allowed to go back to the IDE during the survey.

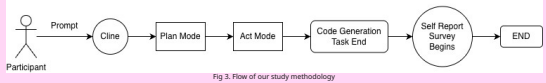


Fig 3. Flow of our study methodology

*our prompt was collected from the DevGPT Dataset: *"Can you write a <programming language of your choice> script/file that checks all the Excel files in the folder and finds the one with "dashboard" sheet. In the dashboard sheet it copies the values from column C to E. Then, it generates another workbook where it copies all the data from the current workbook, and names the new workbook's sheet to whatever the name of the current workbook is."*

Findings

(To better visualize our findings, we divide the interaction between the user and Cline into 3 phases: planning, execution, and evaluation)

Planning Phase

Tasks: user understanding the prompt and requirements, Cline creating a plan for the task, and the user evaluating the plan.

Finding: Software Engineers allocate the most cognitive resources here.

This is due to the importance of tasks in relation to planning and orchestration of the agent to ensure that the it understands the requirements and correctly performs the task.

Execution Phase

Tasks: Cline generating the actual code, running it, and evaluating it.

Finding: Information overload during execution phase

"I'm not reading all of that" -P4. We hypothesize that the real-time generation of text information is too much for the user to comprehend. Even after Cline is done generating a step, they quickly prompt it to proceed to the next. There is minimal engagement here.

Evaluation Phase

Tasks: User evaluates the generated code, its output (if any), and Cline's evaluation.

Finding: Evaluating the output is more important than the process.

Since the output (excel files) were correct, users paid little attention to Cline's overall process, which also included the source code. They failed to properly Recall, Understand, Analyze, and Evaluate certain details about their interaction.

Overall, we find that software engineers' cognitive engagement declines as the task progresses!

Findings (Cont'd)

Software engineers only recalled, understood, analyzed, and evaluated the "Happy Path"

We also find that software engineers do not use their critical thinking abilities to its maximum when recalling, understanding, analyzing, and evaluating Cline's output and process.

Most, if not, all of their cognitive resources are dedicated to the "Happy Path", i.e the path to the correct output. As shown in our survey results, This tunneling vision towards the correct output leads them to ignore potentially critical details of their interaction with Cline.

Design Considerations to make Agentic Coding Assistants more as Tools for Thought

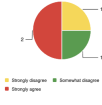
1. **Sustain cognitive engagement by communicating beyond text -** Because the agent is limited to communicating only through text, they are forced to generate significant amounts of information that the user may not fully comprehend. We propose to add other modes of communication such as visualizations [3] and voice capabilities [4] as both have been shown to maintain a user's cognitive engagement.

2. **Use "cognitive forcing designs"-** Because the user fails to think critically beyond the happy path, we propose implementing cognitive forcing designs in the agent. These designs "slow down" [5] the agent's reasoning, which "forces" the user to to engage with the task on their own without over-relying on the agent.

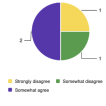
Self Reporting Survey Results

Question	P1 Correct?	P2 Correct?	P3 Correct?	P4 Correct?
Q1 What was the name of the folder you were working on?	Yes	Yes	No	No
Q2 What was the name of the first file created?	Yes	Yes	No	Yes
Q3 How many functions/methods did the generated script have?	No	No	No	No

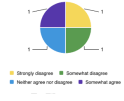
Q5 - "I can reliably provide a concise summary for the first function in the generated script"



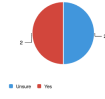
Q6 - "I can reliably provide a concise summary for the last function in the generated script"



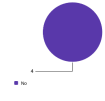
Q7 - "I can reliably determine the order of functions/methods that will be called in the script"



Q8 - The script is able to handle a case when there are no excel files in the working directory



Q9 - Did you make some of your own manual adjustments with the script?



References

[1] Lorin W. Anderson and David R. Krathwohl (Eds.), 2001. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York
[2] Greene, B. A. (2015). Measuring Cognitive Engagement With Self-Report Scales: Reflections From Over 20 Years of Research. Educational Psychologist, 50(1), 14–30. https://doi.org/10.1080/00461520.2014.989230
[3] Matthew O Ward, Georges Grinstein, and Daniel Keim. 2010. Interactive data visualization: foundations, techniques, and applications. AK Peters/CRC Press.
[4] Tze Wei Liew, Su-Mae Tan, Tak Jie Chan, Yang Tian, and Faizan Ahmad. 2025. Cognitive Benefits of Employing Multiple AI Voices as Specialist Virtual Tutors in a Multimedia Learning Environment. Human Behavior and Emerging Technologies 2025, 1 (2025), 8813532
[5] Emily Kuang, Minghao Li, Mingming Fan, and Kristen Shinohara. 2024. Enhancing UX Evaluation Through Collaboration with Conversational AI Assistants: Effects of Proactive Dialogue and Timing. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24).

What would you like to discuss?

We would like to discuss other forms of measuring cognitive engagement in users. While we are familiar with tools such as eye tracking or keypress tracking, we would like to discuss if it is necessary to use brain-computer interfaces (BCIs) in this study. If so, we would like to discuss with the community on best practices on using BCIs.

What would you like to take away from the workshop?

We hope to get critical feedback with our idea. We recognize that developing human-centered AI for collaborative tasks is as much a cognitive, social, neuroscience problem as it is a computational one. We hope to find collaborators in these diverse fields and learn from them. We hope to learn valuable insights on cognitive protection and augmentation, as well as the hardware and software tools researchers in these fields use.

How to proceed with this work/idea?

While we are planning to fully integrate voice and visualization capabilities in Cline, we may encounter some limitations in terms of what we can actually implement in it. Should this be the case, we plan to just pivot to a wizard of oz prototype. We also plan to recruit more participants to participate in this code generation task and survey to gain more insights on their cognitive engagement and propose more novel design considerations.