**Interview Transcript –** Participant: [Anonymized]

**Date of Interview:** 19 February 2025

**Mode of Interview:** Physical

**Interviewer:** Anonymized

**Participant Role:** Software Engineer (Team Lead)

**Duration:** 45:20 mins

**Consent Obtained:** Yes

**Interviewer:** Hello! Thank you for agreeing to participate in the semi-structured interview.

**Participant:**  You’re welcome. It's my pleasure.

**Interviewer:** Today, we are going to discuss code clones and AI-generated code. But before we hop in, I want to ask how many years of experience you have.

**Participant:** I’ve two years of software development experience and I am coding for more than 6years..

**Interviewer:** Great! I am sure you're familiar with LLMs like ChatGPT.

**Participant:** Yeah, I do use ChatGPT and Claude for my day-to-day life. I am very familiar with these tools.

**Interviewer:** Do you use these LLMS to generate code?

**Participant:** Yes, I do take help from LLMs to get the initial structure of my code from these models, and then from there, I carry forward to edit.

**Interviewer:** Did you find anything interesting about the process?

**Participant:**  Basically, if you know what you’re doing, it's a very nice tool that can save a lot of your time for development, but if you don't know what you need to do, then it might increase the time of your development, which also will take longer than it would usually take.

**Interviewer:** Hmm, did you see any repetition or any characteristics by which you can understand that the portion of code is LLM-generated?

**Participant:** I felt LLMS tends to generate code in a very structured manner and very well-documented way. Regarding the repetition, what I experienced was that if you are working with small code snippets, it tends to repeat the same structure, but when the code snippets start getting complex, sometimes it gives a very unique implementation, which is not the normal way to solve the problem.

**Interviewer:** Talking about the repetition, how familiar are you with code clones?

**Participant:** I am familiar with code clones and I know the basics.

**Interviewer:** Do you know the type of clones?

**Participant:** Yes, I do know there are syntactic and semantic clones.

**Interviewer:** I think you have very good idea regarding it, but do you know about the types?

**Participant:** Yes, I do know about the four types of clones and cross-language clones.

**Interviewer:** How do you see cross-language clones in the realm of LLMs? Although cross-language clones are not a part of our study, I'm really interested to know about it.

**Participant:** Okay, basically, sometimes I look for solutions in Stack Overflow, but sometimes I don't get the solution in the programming language I'm looking for. If I get any solution from Stack Overflow, I simply try to translate it to the language I'm working with.

**Interviewer:** That's a very interesting idea. But let's get back to the same language clones. If I understood right, you’re saying you did notice sometimes LLMS gives a very complex solution to your query that is way different than how normally developers would solve the problem?

**Participant:** Not all the time, but it does give a very unique solution.

**Interviewer:** How are those solutions? Does that work properly?

**Participant:** Actually, I won't say it always works, but yes, sometimes it just works fine, but most of the time I need to edit the generated answer.

**Interviewer:** What is your view on detecting AI-generated code? Can you classify which one is AI-generated and which one is not?

**Participant:** Well, I can usually make an educated guess about whether code is AI-generated based on the presence of docstrings and variable names. LLMs like GPT often produce clean, well-documented code. However, that doesn't mean I can say for certain it's AI-generated, as I've seen many developers write exceptionally clean and well-documented code as well.

**Interviewer:** Okay, let me show you some AI-generated and human-written coding examples and see if you can identify which one is AI and which one is human.

**Participant:**

**Demonstration Phase – Summary**

During this phase, the interviewer presented participants with several code snippets: one set written by humans (sourced from Stack Overflow) and another set generated by AI tools, both designed to perform the same functionality. Participants were asked to review the snippets and identify which ones were AI-generated and which were human-written.

**Key Insights:**

* The participant couldn’t determine all the AI-generated code. (after removal of comments)
* The participant consistently identified that LLM-generated code is more structurally consistent and well-documented compared to typical human-written code
* The participant observed that LLMs often generate repetitive patterns in simpler tasks but may produce unconventional or unique implementations for complex problems.
* While the participant could make educated guesses about which code was AI-generated, based on clarity, formatting, and documentation, they acknowledged that these features can also be present in well-written human code. This highlights the detection blind spot for traditional CCD tools, as LLMs blur the boundary between human and machine patterns.
* The participant's reflections, particularly on how prompt framing and task complexity affect code structure, underscore the need for context-aware clone detection approaches that account for the prompt-to-code generation logic of LLMs.