

Surveying the User Requirements of Code Generation AI Models - Perspectives of Model Users and Model Engineers

Thank you for taking the time to consider our study.

Title of Research Study: Code Generation AI Models and User Needs

Principal Investigator, Title, and Affiliation: Professor Dr. Shafay Shamail, Lahore University of Management Sciences

Purpose, Procedure, and Duration: AI based code generation models like Copilot, GPT, Gemini Code Assist, etc. are widely used by coders or software developers and actively maintained and enhanced by model engineers. In this survey we will present some statements and questions that are aimed to better understand the why's and how's of code generation. You will be asked to indicate if you agree with the presented statements and if the presented questions are relevant to you for **better understanding and trusting the code or the model's decision making**. The survey may take about 10 minutes to complete.

Eligibility:

If you are either using some code generation AI models for code reuse or working on training or updating these AI models, your feedback can be very useful to understand how we can improve the models.

Benefits of Study: There is no monetary compensation provided to the participants. The participants can benefit the wider scientific community through their participation and by providing accurate evaluations.

Possible Risks of Study: There are no anticipated risks or adverse effects of this study.

Privacy and Future Use: Your responses to the survey are anonymous. That means we won't know which responses are yours. We won't collect names, internet addresses, email addresses, or any other identifiable information. We may use your responses in future research or share them with other researchers.

Contact:

If you have any questions/clarifications on this study, please contact the Co-Investigator Dr. Shamsa Abid at shamsa.abid@lums.edu.pk

If you have any questions or concerns regarding your rights as a participant in this research study and wish to contact someone unaffiliated with the research team, please contact the LUMS Institutional Review Board Convener at opsteam@lums.edu.pk. When contacting LUMS IRB, please provide the title of the Research Study and the name of the Principal Investigator, or quote the IRB approval number (IRB-0397).

Participant's Declaration

I understand that participation is voluntary. Refusal to participate will involve no penalty.

I declare that I am at least 18 years of age.

I have read and fully understood the contents of this form, and hereby give consent to the Lahore University of Management Sciences research team and its affiliates for this project to collect and/or use my data for the purpose(s) described in this form.

** Indicates required question*

1. Please select "Yes" to indicate you have read this information and you accept to *** take the survey:

Mark only one oval.

☐ Yes

Demographic data

2. Please indicate your education subject area ***

Mark only one oval.

☐ Computer Science

☐ Software Engineering

☐ Artificial Intelligence

☐ Data Science

☐ Computer Engineering

☐ Electrical Engineering

☐ Mathematics

☐ Other: _____

3. Please indicate your current education level *

Mark only one oval.

- ☐ Undergraduate
- ☐ Bachelors Degree
- ☐ Enrolled in Masters
- ☐ Masters Degree
- ☐ Enrolled in PhD
- ☐ PhD Degree
- ☐ Other: _____

4. Please Indicate your current role *

Tick all that apply.

- ☐ Student
- ☐ Researcher
- ☐ Software Engineer/ Developer/ Programmer
- ☐ Software Quality Engineer/ Tester
- ☐ ML Engineer/ AI Specialist
- ☐ Assistant/ Associate/Full Professor
- ☐ Other: _____

5. Please indicate the time period during which you have been working with AI tools/models for code generation. *

Mark only one oval.

- ☐ Less than 2 months
- ☐ 2 to < 6 months
- ☐ 6 to < 12 months
- ☐ 1 to < 2 years
- ☐ 2 to < 5 years
- ☐ 5 years or more

6. Which AI code generation models do you use or interact with *

Tick all that apply.

- ☐ GitHub Copilot
- ☐ Tabnine
- ☐ Codeium
- ☐ Amazon CodeWhisperer
- ☐ Replit Ghostwriter
- ☐ Cursor
- ☐ ChatGPT (GPT-4, GPT-4o, etc.)
- ☐ Claude (Claude 3.5)
- ☐ Google Gemini
- ☐ Mistral AI / Codestral
- ☐ Meta Code Llama
- ☐ StarCoder / StarCoder2
- ☐ PolyCoder
- ☐ SantaCoder
- ☐ DeepSeek Coder
- ☐ IntelliCode
- ☐ JetBrains AI Assistant
- ☐ Other: _____

7. Please indicate your interaction with code generation models *

Mark only one oval.

- ☐ I use AI code generation tools for writing code only *Skip to question 16*
- ☐ I develop or train AI models for code generation
- ☐ I am both using and engineering AI code generation models
- ☐ I use and evaluate the code generation models for trust and transparency
- ☐ Other: _____

Code Generation AI Model Requirements: Model Engineer Perspective

Imagine you are creating/engineering/enhancing an AI model for code generation. Apart from the generated code, you may need more explanation of how and why the output code is generated **for better understanding the model and its decision-making**.

Please indicate your level of agreement with the following capabilities that code generation models can aim to provide in addition to generating code. (1=strongly disagree, 2= disagree, 3= neutral, 4= agree, 5= strongly agree)

8. 1) Understanding the attention mechanisms or similar mechanisms used by the model to focus on relevant parts of the input data during code generation can provide insights into how the model makes decisions. *

Rationale: Visualizing attention weights or attention maps can help me understand which parts of the input data are most influential in generating specific code tokens

Mark only one oval.

1	2	3	4	5	
<hr/>					
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree
<hr/>					

9. 2) Visualizing the internal representation of the code structure learned by the model, such as abstract syntax trees (ASTs) or code embeddings, can provide insights into how the model captures the hierarchical structure and semantic relationships of code snippets. *

Rationale: This visualization can aid in debugging model architecture issues and understanding the learned representations.

Mark only one oval.

1	2	3	4	5	
<hr/>					
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree
<hr/>					

10. 3) Logging the generation process, including intermediate steps, decisions, and probabilities associated with each generated token, allows me to trace the model's behavior and diagnose performance issues. *

Rationale: Analyzing the generation log can help identify patterns, biases, or errors in the model's output and guide model refinement efforts

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

11. 4) Identifying which input features or representations are most influential in generating specific code tokens or structures can help me understand the factors driving code generation decisions. *

Rationale: Feature importance analysis can guide feature engineering, input preprocessing, and model optimization strategies to enhance the model's performance.

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

12. 5) Analyzing errors made by the model, such as incorrect predictions, syntactic errors, or semantic inconsistencies in the generated code, helps me diagnose model weaknesses and identify areas for improvement. *

Rationale: Understanding the types and patterns of errors made by the model can inform targeted error correction techniques and model updates.

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

13. 6) Incorporating domain-specific knowledge or rules into the model interpretation process can provide insights into how the model performs in specific programming languages, software engineering contexts, or application domains. *

Rationale: This domain-aware interpretation can inform model refinement and customization for different use cases and scenarios.

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

14. 7) Developing tools and utilities for debugging the code generation model, such as techniques for analyzing model gradients, inspecting internal model states, and visualizing model predictions, can streamline the model development process and accelerate troubleshooting efforts. *

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

15. Please add any other thing you may need to better understand and trust the model's decision making

Code Generation AI Model Requirements: Coder or Software Developer Perspective

Imagine you are using an AI model for code generation. Apart from the generated code, you may need more explanation of the output code **for better understanding and trusting the code**.

Please indicate your level of agreement with the following capabilities that code generation models can aim to provide in addition to generating code. (1=strongly disagree, 2= disagree, 3= neutral, 4= agree, 5= strongly agree)

16. 1) Provide an explanation of the overall structure and organization of the generated code, including the purpose of each component, the flow of control, and the relationships between different modules or functions. *

Rationale: To help understand the high-level design decisions made by the AI model

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

17. 2) Explain the underlying code patterns or design principles used by the AI model to generate code snippets. *

Rationale: To help understand the reasoning behind the code generation process. This could include insights into common programming idioms, best practices, and architectural patterns reflected in the generated code.

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

18. 3) Provide contextual information about the input data, such as the input specifications, requirements, constraints, or user intentions *

Rationale: To help understand the context in which the code was generated and interpret its relevance and suitability for the given task or problem

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

19. 4) Offer explanations for variable names, function names, and comments included in the generated code. *

Rationale: To provide insights into the semantics and intentions behind the code constructs. Clear and meaningful naming conventions and descriptive comments enhance code readability and facilitate comprehension

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

20. 5) Explain the dependencies and external libraries used by the generated code, as well as their purpose and relevance to the task at hand. *

Rationale: To help understand the external resources required for the code to function properly and assess any potential dependencies or integration issues

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

21. 6) Present alternative code options or variations generated by the AI model, along with explanations for why each option was considered and the trade-offs involved. *

Rationale: To explore different implementation choices and make informed decisions based on the provided insights

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

22. 7) Provide estimates of model confidence or uncertainty associated with the generated code, as well as any limitations or caveats associated with the predictions. *

Rationale: To help assess the reliability and trustworthiness of the generated code and make informed judgments about its suitability for deployment.

Mark only one oval.

	1	2	3	4	5	
disa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	agree

23. Please add any other thing you may need to better understand and trust the code

Questions about Code Generation AI Models: General Perspective

Please indicate if the following questions are relevant for **better understanding and trusting the code or the model's decision making**. (1=strongly irrelevant, 2= not relevant, 3= neutral, 4= relevant , 5= strongly relevant)

24. 1) How does the AI infer user intent from ambiguous or incomplete prompts? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

25. 2) Can I receive feedback on missing or unclear input specifications before generation? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

26. 3) What assumptions does the model make when default values or parameters are not specified? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

27. 4) Does the generated code follow current best practices in the target language? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

28. 5) Is the generated code idiomatic and maintainable according to human standards? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

29. 6) Does the AI generate unit tests or suggest test coverage for the code? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

30. 7) How does the AI ensure compatibility of generated code with modern dependencies? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

31. 8) What internal representations (e.g., embeddings, ASTs) are used during code generation? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

32. 9) How does the model handle domain-specific logic during code generation? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

33. 10) How does the accuracy of the model vary with different prompt styles or lengths? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

34. 11) Does model confidence correlate with actual correctness of the output code? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

35. 12) How scalable is the model for large and complex software tasks? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

36. 13) How can I guide the model to use specific algorithms or patterns? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

37. 14) Can I influence the level of abstraction or granularity in generated code? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

38. 15) Can I prioritize runtime efficiency vs. readability in the output? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

39. 16) Is there a way to lock parts of the prompt so the model respects certain constraints? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

40. 17) Why did the model choose one library or API over another? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

41. 18) Why was a specific data structure selected in the generated solution? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

42. 19) Can I audit or inspect the type and source of training data relevant to my prompt? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

43. 20) How recent is the training data the model uses for its predictions? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

44. 21) What specific code domains or languages does the model struggle with? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

45. 22) What types of bugs or errors is the model most prone to introducing? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

46. 23) What happens if I input code that mixes multiple programming paradigms? *

Mark only one oval.

1 2 3 4 5

not ☐ ☐ ☐ ☐ ☐ strongly relevant

47. 24) How does the model behave when code with logical contradictions is presented? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

48. 25) What are the computational costs of using this model for code generation at scale? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

49. 26) How can I integrate this AI tool within my existing software development toolchain? *

Mark only one oval.

1 2 3 4 5

not relevant ☐ ☐ ☐ ☐ ☐ strongly relevant

Thanks for your feedback!

We would like to know your thoughts on this survey!

50. Any thoughts.....

This content is neither created nor endorsed by Google.

Google Forms

