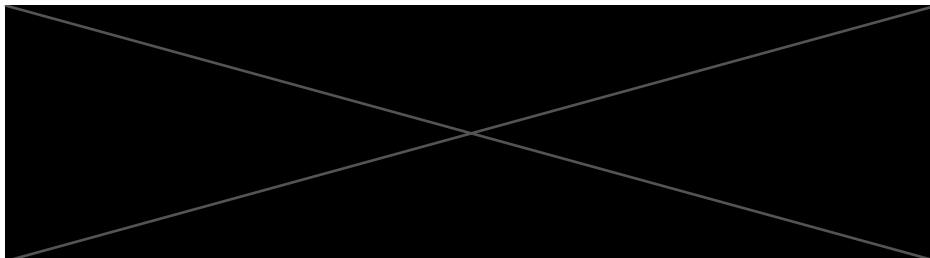


Introduction to the Survey

Who we are.



We are a team of software engineering researchers investigating the use of Artificial Intelligence in software documentation. Our work focuses on understanding how Large Language Models (LLMs) can automatically generate architectural documentation and how developers perceive and evaluate these outputs.

The main goal of this study.

The goal of this study is to explore how developers assess the *value, comprehensibility, accuracy, and limitations* of LLM-generated architectural documentation. We aim to better understand whether this type of automatically produced documentation can support software engineering practices and improve documentation quality in real-world projects.

How your data will be processed.

Any personally identifiable information will be removed before analysis and will not appear in any publication. Your responses will be anonymized, and all raw data containing identifiable information will be securely deleted once the analysis is complete and the study results have been finalized.

Who will have access to the data.

Only the members of the research team will have access to the collected data. An anonymized and aggregated version of the dataset may be shared in future scientific publications or research repositories to ensure transparency and replicability.

Can you withdraw your consent?

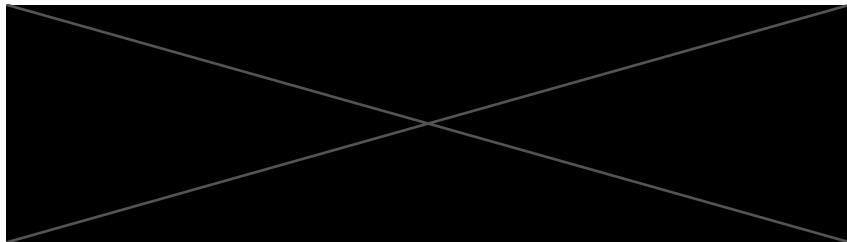
Yes. You can withdraw your participation at any time during the process. Upon request, all data associated with you will be deleted immediately and excluded from any analysis or publication.

Ethical aspects.

The ethical impact of this research has been reviewed. No personal, sensitive, or potentially harmful information is collected. Participation is voluntary, and no risks or direct benefits are expected.

Contact.

If you have any questions about this research or how your data will be used, please contact the research team at:



The data will be collected and processed for the sole purpose of this research in compliance of GDPR.



1. Email *

2. What is your current job? *



- Software Developer
 - Researcher
 - Phd student
 - _____
-

3. How many years of experience do you have in software development? *



- 0-1
- 2-5
- more than 5 years



4. I'm familiar with software architecture concepts (e.g., UML, Architectural Pattern, Architectural Style)



1 2 3 4 5

Strongly Agree

Strongly

5. What has been your role in this project? *



Contributor

Maintainer

Main Developer

6. How would you classify the project this repository comes from? *



Research / Academic project

Industrial / Professional project

Personal project

Open-source community project



7. Please describe the application domain of the repository (e.g., web application, data processing, DevOps tooling, machine learning, IoT, automotive, healthcare, finance, etc.)

8. Before reading the documentation, I was familiar with the codebase of this project. *



1 2 3 4 5

Strongly Agree

9. Have you ever written or maintained documentation for this repository (e.g., README, architecture docs, wiki, ADRs)?



Yes

No

10. If yes, please briefly describe the type of documentation you worked on. *



Survey

11. The LLM-generated architectural documentation provides valuable insights into the architecture of my repository.



1 2 3 4 5

Strongly agree

12. The LLM-generated documentation is useful for understanding and maintaining the system's structure and dependencies.



1 2 3 4 5

Strongly agree

13. The LLM-generated documentation is valuable enough that I would consider integrating it into my project (e.g., as a README or ARCHITECTURE.md).



1 2 3 4 5

Strongly agree



14. Which sections or features of the LLM-generated documentation (e.g., Context, Containers, Diagrams, Code-Level View) did you find most valuable, and why?

15. The LLM-generated architectural documentation is clear, well-structured, and easy to follow.



1 2 3 4 5

Stro Strongly agree

16. The LLM-generated architectural documentation uses appropriate terminology for software architecture.



1 2 3 4 5

Stro Strongly agree



17. The LLM-generated architectural documentation contains excessive redundancy or unnecessary information.



1 2 3 4 5

Stro Strongly agree

18. The explanations within the documentation are sufficiently detailed *



1 2 3 4 5

Stro Strongly agree

19. *Which parts of the documentation were most or least comprehensible to you? **

20. The LLM-generated documentation accurately reflects the architecture implemented in the repository.



1 2 3 4 5

Stro Strongly agree



21. The documentation is internally consistent across sections and diagrams. *



1 2 3 4 5

Stro Strongly agree

22. The content is complete and does not omit key architectural elements. *



1 2 3 4 5

Stro Strongly agree

23. The relationships and dependencies among components are well captured *



1 2 3 4 5

Stro Strongly agree

24. The component responsibilities are correctly described in the documentation *



1 2 3 4 5

Stro Strongly agree



25. I would use the documentation as a reliable reference for the system architecture *



1 2 3 4 5

Stro Strongly agree

26. Section 1 – System Overview: The purpose, scope, and key features described are correctly derived from the repository content.



1 2 3 4 5

Stro Strongly agree

27. Section 2 – Architectural Context: The external systems, APIs, data sources, and actors are correctly derived from the repository content.



1 2 3 4 5

Stro Strongly agree

28. Section 2.1 – Use Case Diagram: The diagram correctly represents actors and use cases derived from the repository content.



1 2 3 4 5

Stro Strongly agree

29. Section 3 – Containers: The described containers, their responsibilities, technologies, and communication protocols match the repository content.



1 2 3 4 5

Stro Strongly agree

30. Section 3.1 – Component Diagram: The diagram correctly reflects the containers and externals as found in the repository.



1 2 3 4 5

Stro Strongly agree

31. Section 4 – Components: The identified components, their responsibilities, and interactions correspond to code-level relationships (imports, calls, dependencies).



1 2 3 4 5

Stro Strongly agree

32. Section 5 – Code-Level View: The entry points, key modules, and patterns correspond to actual code structure (no speculative elements).



1 2 3 4 5

Stro Strongly agree



33. Section 5.1 – Code-Level View: The diagram includes only elements and relationships explicitly present in the repository



1 2 3 4 5

Stro Strongly agree

34. Section 6 – Cross-Cutting Concerns: The concerns listed (e.g., security, logging, configuration, testing) are correctly represented (no speculative elements).



1 2 3 4 5

Stro Strongly agree

35. Section 7 – Quality Attributes & Rationale: The quality attributes and rationales are correctly represented (no speculative elements).



1 2 3 4 5

Stro Strongly agree

36. Section 8 – Deployment View: The deployment information (nodes, environments, dependencies) accurately reflects the repository's infrastructure configuration.



1 2 3 4 5

Stro Strongly agree



37. Section 8.1 – Deployment Diagram: The diagram includes only infrastructure elements and connections declared in the repository.



1 2 3 4 5

Strongly agree

38. Please describe any inconsistencies, inaccuracies, or unclear parts you noticed in the documentation. Which sections need improvement?

39. The LLM-generated documentation completely represents the system's architecture without missing relevant views or elements.



1 2 3 4 5

Strongly agree

40. The diagrams and textual sections are free from speculative or invented information.*



1 2 3 4 5

Strongly agree



41. The level of abstraction used in each section (e.g., Context, Containers, Components) is appropriate for the project.



1 2 3 4 5

Stro Strongly agree

42. The overall structure of the documentation (sections 1–8.1) fits well with my project's architecture.



1 2 3 4 5

Stro Strongly agree

43. What important architectural information is missing or insufficiently represented in the LLM-generated documentation?

44. What improvements or changes would make the LLM-generated documentation more useful and trustworthy for your project?

