Vibe Coding A Research Probe for Exploring AI/Voice Based Code Reviews

Martin Gundtoft
Copenhagen School of Design and Technology, Denmark

marg@ek.dk

Abstract: Generative AI tools increasingly shape established software engineering practices such as code review, but the socio-technical implications of using AI for these practices remain understudied. In this paper we first introduce vibe coding (Andrej Karpathy [@karpathy], 2025) as a method for allowing researchers with limited coding experience to rapidly create custom made probes for conducting research. Guided by Alami and Ernst's (2025) findings on AI-generated feedback for code review, we introduce a vibe coded AI/Voice based code review prototype as a provotype (Boer and Donovan, 2012). We then outline an explorative study to critically assess the socio-technical effects of using AI based voice interfaces in code reviews. We propose a qualitative approach, based on the Disruptive Research Playbook (Storey *et al.*, 2024), involving Danish software developers to investigate voice-based feedback's impact on topics including trust, collaboration, and perceived skill shifts. Initial methodological reflections emphasize the need for cautious exploration using the provotype as an intervention for gathering data in the form of reactions, expectations, and concern about the effects of AI interactions, in the established professional practise of code review. Next steps are to finalize the provotype, complete the research design and collect and analyze qualitative data from interventions with danish software developer teams.

Keywords: AI-assisted code review, Vibe coding, Generative AI, provotypes, Socio-Technical studies, Human-AI collaboration

1. Introduction

The field of software development has adopted generative AI tools such as GitHub Copilot and DeepCode into tasks like debugging, testing, and code review (*AI Code Review | IBM*, 2024). However, the socio-technical implications, particularly how voice-based AI might alter developer interactions and feedback processes, remain underexplored. Existing studies indicate nuanced differences in developer engagement with AI versus human-generated feedback, highlighting trust and cognitive load issues (Alami and Ernst, 2025).

Addressing this gap, our research employs vibe coding - utilizing generative AI to create code through natural language prompts - to rapidly prototype a voice-based AI code review tool. Vibe coding significantly lowers prototyping barriers, making sophisticated coding tools accessible to researchers without extensive technical backgrounds (*What is Vibe Coding? | IBM*, 2025).

We introduce this tool as a provotype, a deliberately provocative prototype designed to elicit deeper user insights(Boer and Donovan, 2012), guided by the Disruptive Research Playbook (Storey *et al.*, 2024). Our qualitative study involves Danish software developers, and explores how an intervention in the form of code review feedback given by an AI/voice based provotype influences collaboration, trust dynamics, and perceptions of skill enhancement or erosion.

With the research proposed in this WIP paper, we aim to (1) provide valuable insights in the viability of using vibe coding as a method for low-ressource AI-driven prototyping in research, and (2) add to the body of knowledge that critically assess generative AI's role in software development practises, with early insights for the software-engineering community.

2. Background

Code review remains a central practice for ensuring software quality, traditionally involving collaborative, interpersonal communication. AI-assisted reviews introduce new dynamics, prompting exploration into their socio-technical impact. Research by Alami and Ernst (2025) indicate that developers process AI feedback similarly to human feedback but face challenges in other areas, including trust and cognitive load.

Vibe coding, an emerging practice leveraging generative AI for rapid software prototyping, allows researchers without extensive coding experience to quickly implement advanced interfaces (*What is Vibe Coding? | IBM*, 2025). This approach lowers the barrier for prototype development, essential for exploring innovative AI interactions.

The concept of provotypes - prototypes specifically designed to provoke user reflection - is valuable in contexts where implications of technology are uncertain (Boer and Donovan, 2012). Combining vibe coding and

provotyping enables researchers to quickly deploy provocative AI tools, eliciting critical insights into user responses and potential disruptions.

Storey et al.'s (2024) Disruptive Research Playbook provides methodological guidance, advocating for a socio-technical perspective to understand long-term implications of disruptive technologies. This approach encourages human-centered inquiry into how emerging technologies might alter established professional practices.

Our research propose to integrate these theoretical and methodological frameworks to investigate how voice-based AI code reviews could influence developer collaboration, trust formation, and perceived skill shifts, contributing to broader understandings of AI's socio-technical impacts.

No prior work, to our knowledge, has examined this, making it an open question in human/AI code collaboration. Similarly, little has been written about how researchers can rapidly prototype AI experiences (vibe coding) to study such questions.

3. Methodology

The provotype was developed using Cursor, an AI-powered, agentic coding tool powered by the most prominent foundation models, and GPT-4 for assisting in conceptual generative tasks. This method streamlined development, translating natural-language instructions into functional React code, integrated with speech recognition and speech synthesis through the Gemini live API. The code-reviewer was repurposed from a voice-based AI prototype for simulating exams, developed by and educator in BACs software development program. While the researcher who developed the provotype possessed some knowledge of coding and software development, recent advancements in the available tools indicate a decreasing reliance on domain-specific expertise.

In a provotype code review session, the user initiates the review process through a web interface, with optional configuration for expertise level, specific area of interest and more. A dialogue based session, where the AI has access to the users shared screen (and a Github Repo if permitted), is then launched. The AI has an elaborate set of instructions based on best practices on how to behave, analyze and give feedback in a code review context. During the session, the AI generates structured, spoken feedback highlighting issues, recommending improvements, and addressing clarifying questions in a responsive, conversational format. The suggestions are logged via a dynamic, text-based interface. After the session, the user is presented with a summary of the session, including a transcript and recommendations (Gundtoft, 2025).

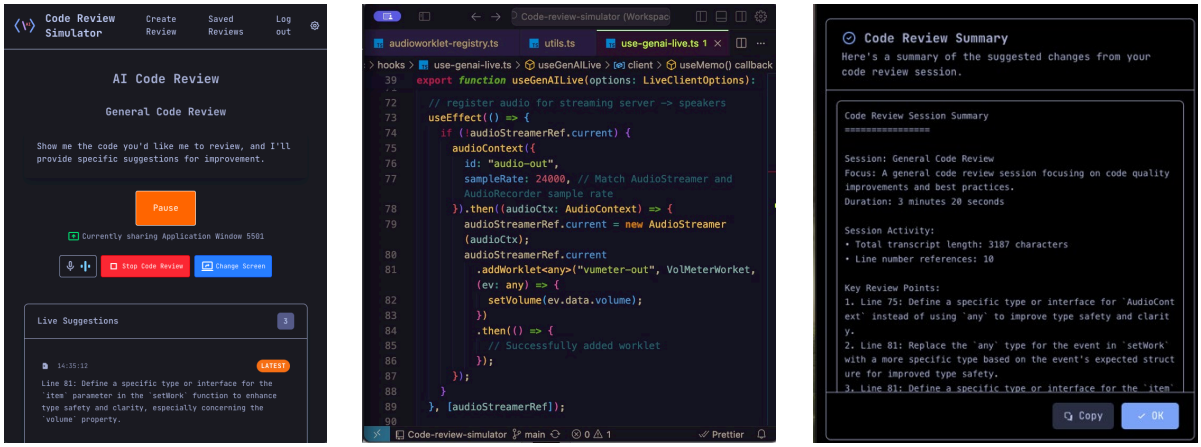

Figure 1: Left: User interface with suggestions. Middle: Code being reviewed. Right: Session summary

Development proceeded iteratively, guided by continuous testing and evaluation with an in-house software developer. Initial prototypes produced overly verbose and unfocused feedback, leading to iterative prompt and code adjustments that ensured concise, comprehensible audio feedback suitable for practical use.

This development process illustrates vibe coding's potential to lower the barrier for creating custom made research probes, empowering researchers without extensive coding background to prototype advanced human-computer interactions. However, reliance on AI-generated code required careful human oversight to identify and correct errors or misinterpretations, emphasizing cautious and critical engagement, and mirroring broader concerns in AI-driven development.

The purpose of the probe was not to showcase a proof-of-concept for voice-based code review though, but rather to act as a provotype - a conversational artifact we can take into the field to probe software engineers' interactions with AI. Next, we describe our preliminary research design leveraging this provotype to explore AI-assisted code reviewing in situ.

4. Research Design

In the study we intend to employ the prototype as a provotype within a qualitative research framework informed by the Disruptive Research Playbook (Storey *et al.*, 2024). The guiding research question of our exploratory study asks: How do software developers experience and envision the prospect of using AI feedback for established practices like code review, with a special emphasis on speech enabled AI in areas like collaboration, and perceptions of skill development/erosion? Given the exploratory and experiential nature of the study, findings are expected to mainly yield new investigative insights, not definitive answers to the research question or predictions of actual future deployment outcomes.

We plan to engage 2-3 Danish software development teams (approximately 4–5 members each). While Denmark's open, collaborative professional culture provides a suitable context for this exploratory investigation, it also aims to offer insights relevant to research in other cultural settings, with adaptability to varying contexts and scales. Each session will last approximately 75 minutes, structured in three phases:

- **Introductory Phase (15 min):** Briefing on prototype functionalities and informed consent procedures.
- **Interaction Phase (30 min):** Team members engage in sessions with the provotype, and have snippets of their own code reviewed. Participants interact freely with the AI, posing follow-up questions and discussing feedback among themselves, while the reseachers obeserve and take notes. Before the study, we aim to update the provotype with audio- and screen-recording functionality.
- **Reflective Phase (30 min):** Semi-structured group interviews gather participant reflections on usability, perceived trustworthiness, emotional responses, and comparative insights versus traditional review practices.

Data collection combines observational data, interaction logs, and interview transcripts. Thematic analysis, will be used to identify emerging sociotechnical dynamics.

Validity considerations include triangulating self-reported reflections against observational data. We acknowledge the artificial nature of the provotype setup, noting that initial reactions may differ from sustained use scenarios. Ethical considerations include explicit participant consent, transparent discussion of AI limitations, and secure data handling.

This research design enables us to explore immediate and latent sociotechnical responses to a provocative AI tool, laying groundwork for future studies on sustained use and broader adoption implications.

5. Discussion

Our approach aims to deliver two significant contributions: (1) demonstrating vibe coding's methodological potential for creating rapid research probes and (2) findings that critically assess generative AI's role in software development practises.

Initial reflections anticipate several potential socio-technical shifts. Voice-based AI feedback could alter emotional and cognitive engagement, make interactions feel more personal, or erode established social practices and relationships with colleagues. These shifts may affect trust, potentially alleviating or intensifying issues identified by Alami and Ernst (2025), depending on feedback realism and accuracy.

Methodologically, vibe coding's accessibility introduces a paradox: while lowering barriers enables valuable exploratory research, reliance on generative AI necessitates oversight to ensure scholarly rigor and methodological clarity. Researchers must critically reflect on AI's role in their processes, paralleling broader debates on human oversight in AI-driven automation.

Future research should explore longitudinal impacts, cross-cultural differences, and larger participant samples. Additionally, integrating quantitative measures could further validate qualitative insights.

6. Conclusion

This WIP paper has presented an approach combining vibe coding and provotyping to explore socio-technical implications of voice-based AI code review. Our proposed exploratory study offers a practical research

framework to investigate AI's nuanced impacts on software practices. Methodological reflections highlight the necessity of cautious and critical engagement with generative AI. This work provides foundational insights into responsible, human-centered AI research practices, supporting future studies and informing broader discussions on AI's role in professional software development contexts.

Acknowledgements

The author gratefully acknowledges Benjamin Hughes for creating the exam simulator prototype that was used as a basis for vibe coding the code reviewer.

Ethics declaration: This research is in the planning phase and has not yet commenced data collection. Ethical approval will be sought from the informants prior to conducting participant studies.

AI declaration: The paper was developed using AI tools for specific purposes. The prototype discussed was created using the AI coding assistant Cursor and OpenAI's GPT-4 for generating React code and refining system prompts. Additionally, GPT-4 was used to support early drafting and outlining of this paper, particularly for improving structure, grammar, and flow.

References

AI Code Review | IBM (2024). Available at: https://www.ibm.com/think/insights/ai-code-review (Accessed: 11 July 2025).

Alami, A. and Ernst, N. (2025) 'Human and Machine: How Software Engineers Perceive and Engage with AI-Assisted Code Reviews Compared to Their Peers', in *2025 IEEE/ACM 18th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*. *2025 IEEE/ACM 18th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE)*, Ottawa, ON, Canada: IEEE, pp. 63–74. Available at: https://doi.org/10.1109/chase66643.2025.00016.

Andrej Karpathy [@karpathy] (2025) 'There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper', *Twitter*. Available at: https://x.com/karpathy/status/1886192184808149383 (Accessed: 11 July 2025).

Boer, L. and Donovan, J. (2012) 'Provotypes for participatory innovation', in *Proceedings of the Designing Interactive Systems Conference*. *DIS '12: Designing Interactive Systems Conference 2012*, Newcastle Upon Tyne United Kingdom: ACM, pp. 388–397. Available at: https://doi.org/10.1145/2317956.2318014.

Gundtoft, M. (2025) *Code Review Simulator*. Available at: https://keamarg.github.io/Code-review-simulator/ (Accessed: 11 July 2025).

Storey, M.-A. *et al.* (2024) 'A Disruptive Research Playbook for Studying Disruptive Innovations'. arXiv. Available at: https://doi.org/10.48550/ARXIV.2402.13329.

What is Vibe Coding? | IBM (2025). Available at: https://www.ibm.com/think/topics/vibe-coding (Accessed: 11 July 2025).