

BCS Kent

Vibe Coding - Q&A

Thank you to all those who attended my Vibe Coding presentation on Thursday 22nd January 2026, we had a lot of questions during the event (which is fantastic), I would like to use this document to provide answers to those questions posted online, that I have not previously answered.

If you wish to connect / chat or provide additional feedback, you can reach me on LinkedIn
> <https://www.linkedin.com/in/davpea/>

All answers reflect my personal views / comments only.

Regards

David Pearson MBCS CEng

Questions & Answers

Q: For beginners wanting to try which are the best cheap / free versions

A: For complete beginners who may not even want to learn to code or see the code being generated - so you just want a finished / running app, popular choices are [Base44.com](https://base44.com), replit.com or lovable.dev

Q: Can you vibe code for integration between the different modules/units you are creating?

A: Yes, you would simply create a prompt containing a description of the integration required. If you are working with specific middleware or security stacks, either add details of these to the prompt to enable download and/or specify specifics in the prompt .. think about what the integration looks like / how to describe in English and you have a path to success 

Q: Is it fair to say that coders are still required, as developing prompts is coding but at a higher level, using well structured natural language?

A: Coders are still required .. I see it as an evolution, apart from understanding the coding language you now have tools available to enable and augment your success, learning how to use the AI coding assistants is how you differentiate from other developers.

Q: Which developer environment are you using?

A: For the first demo (Python), I use [VSCode](https://code.visualstudio.com) with the [OpenAI extension](https://openai.com). I decided on this choice due to the popularity of VSCode. For the second demo (SwiftUI mobile app), I use Apple XCode which has Apple intelligence built in, this allows a ChatGPT (Code) or Claude Code account to be specified and used. Don't forget you need an OpenAI / ChatGPT or Claude Code account 😊

Q:Impressive, but aren't there issues related to code maintenance?

A: As you are aware, code maintenance is different from initial code generation. You can of course use AI Assistants to help with maintenance, one technique is to ensure the code is documented, so prompts can target blocks of code, also sometimes API's of functions being used from a core language change how they work etc ... I create a small number of prompts to access these on a regular basis eg. Analyse the code against language documentation to find differences.

Q:Is there a performance benefit in giving the whole prompt once? Would a “dialogue” produce a different / faster result?

A: The choice between a large and short prompt for coding depends heavily on the complexity of the task, with [recent research](#) suggesting that while long prompts provide necessary context for complex systems, shorter, more concise prompts often yield more accurate code for specific functions. The consensus is that clarity and structure matter more than raw length.

Q:Did you experiment with running models locally rather than in the cloud? Consider IP-sensitive codes

A: Yes - both hugging face and Ollama enable download / use of local LLM's ... You can intercat at the command line or via your IDE, and works very well for specific use cases eg. legal or health. I would use the cloud based LLM's as default though when using Codex or Claude.

Q:How maintainable is the code produced? If you want to go back to the app to add extra features. What constraints are there? how repeatable is the code

A: See my earlier answer on this.

Q:BTW your linkedIn address on the first slide is wrong.

A: Thanks for pointing that out - I've updated it now on the slides and also added to the start of this document.

Q:How to reference and draw on datasets held outside the App?

A: Use RAG or look at [MCP](#) / Agentic AI.

Q:Is AI code generators used quite widely - are there statistics on code generator usage in production environments?

A: The use of AI code assistants is growing at a rapid pace, there or no official statistics I'm aware of yet 😊

Q:How easy to building standard monitoring agents into the code to be able to monitor condition of app from support services?

A: I would look at creating test cases and then via agentic AI, have them run in the background and report any issues.