

## Hey! What's New? 2025-83

### Programming Tasks You Shouldn't Hand Off to AI

David Gewirtz writes, in an article in *ZedNet*, that “despite the fact that tech companies like Microsoft are showing coders the door by the thousands, AI cannot, and will not, be the sole producer of code. In fact, there are many programming tasks for which an AI is not suited.” This article focuses on programming tasks where you shouldn't use an AI.

**Complex systems and high-level design:** Generative AI systems are essentially super-smart auto-complete, Gewirtz says. “They can suggest syntax, they can code, and they can act as if they understand concepts. But all of that is based on probabilistic algorithms and a ton of information scraped from the web. Contextual intelligence is not a strength. Just try talking to an AI for a while, and you'll see them lose the thread. If you need to produce something that requires substantial understanding of how systems interact, experience to make judgment calls about trade-offs, understanding of what works for your unique needs, and consideration of how everything fits with your goals and constraints, don't hire an AI.”

**Proprietary codebases and migrations:** “Large language models are trained on public repositories and (shudder) Stack Overflow. More than likely, if you give an AI your proprietary code and ask it to do big things, you'll embed many lines of plausible-looking code that just won't work. I find that using AI to write smaller snippets of code that I otherwise would have to look up from public sources can save a huge amount of time. But don't delegate your unique value add to a brainy mimeograph machine.”

**Innovative new stuff:** Don't try to get an AI to be an innovator, Gewirtz warns. “AIs can do wonders with making boilerplate look innovative, but if you need real out-of-the-box thinking, don't use a glorified box with brains. This applies not only to functional coding, but to design as well. To be fair, AIs can do some wonderful design. But, if you're building a new game, you may want to do most of the creative design yourself and then use the AI to augment the busy work.”

**Critical security programming and auditing:** Do not let the fox guard the hen house, Gewirtz advises. “Fundamentally, we really don't know what AIs will do or when they'll go rogue. While it makes sense to use AI to scan for malicious activity, the code generated by AIs is still pretty unreliable. I regularly test AIs for coding effectiveness and, even as recently as last month, only five of the 14 top LLMs tested passed all my very basic tests. Let AIs help you out. But don't trust an AI with anything really important.”

**Domain-specific business logic:** Asking an AI to write code about your unique business operations is a recipe for failure, Gewirtz warns. “Keep in mind that AIs are trained on a lot of public knowledge. But the AIs are not trained on your internal business knowledge, trade secrets, practices, folklore, long-held work-arounds, yada yada yada. Use the AI for what it's good at, but don't try to convince it to do something it doesn't know how to do.”

**Low-level systems work and performance optimizations:** While it's possible for an AI to identify areas of code that could use optimization, he notes, “there are limits. AIs aren't trained on the

very fine details of microarchitectural constraints, nor do they have the experience of coaxing just a skosh more out of every line of code. If you need fine craftsmanship, you'll need a fine craftsman -- in this case, a very experienced coder."

**Learning exercises and educational assignments:** "If you're a student or an educator, AI is a boon. But be careful. Don't cheat, and don't use it to shortcut work that you really should be doing to make education happen. But consider how it might help augment your studies or help you keep up with students' demands."

**Collaboration and people stuff:** Gewirtz has found that "if I treat the AI chatbot as if it were another human coder at the other end of a Slack conversation, I can get a lot out of that level of 'collaboration.' A lot, but not everything. Both humans and AIs can get stubborn, stupid and frustrating during a long, unproductive conversation. Humans can usually break out of it and be persuaded to be helpful, at least in professional settings. But once you reach the limit of the AI's session capacity or knowledge, it just becomes a waste of time."

For considerable more, please see [9 programming tasks you shouldn't hand off to AI - and why | ZDNET](#).