

MY WEIRD PROMPTS

Podcast Transcript

EPISODE #104

Vibe Coding & The Rise of the AI Orchestrator

Published December 25, 2025 • Runtime: 24:15

<https://myweirdprompts.com/episode/vibe-coding-agentic-development/>

EPISODE SYNOPSIS

Are we witnessing the end of the traditional programmer? In this episode of My Weird Prompts, Herman and Corn dive into the world of agentic development and "vibe coding," exploring how tools like Claude Code are shifting the focus from syntax to systems thinking. They discuss how the role of the developer is evolving into that of an "orchestrator," where managing AI agents is more critical than memorizing semicolons. Whether you're a seasoned dev or a tech-curious problem solver, learn why the ability to plan and manage complex systems is the most valuable skill for the year 2026.

DANIEL'S PROMPT

Daniel

I'd love to discuss how technology roles are being redefined in the era of AI code generation. As someone who is more interested in problem-solving and making computers "do things" than in writing code, I see AI tools as powerful upskilling catalysts. Using agentic tools for development relies more on planning, task management, and "thinking" than traditional programming. Where do people like me fit into the tech teams and HR structures of the future? How will we work alongside human developers and AI to build software in today's and tomorrow's businesses?

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts! I am Corn, and I am feeling especially relaxed today here in our sunny Jerusalem living room. I am joined, as always, by my much more high energy brother.

Herman

Herman Poppleberry at your service! And yes, I have had my morning oats and I am ready to dive into some serious tech talk. Being a donkey has its perks, mostly in the stamina department, which I am going to need because our housemate Daniel sent us a really deep one today.

Corn

Yeah, Daniel was telling me about this while we were making coffee this morning. It is a topic that hits close to home for anyone who likes to tinker with computers but maybe never felt like a hardcore programmer. I am a sloth, naturally, so I am all about finding the most efficient way to get from point A to point B without doing unnecessary work. And it sounds like technology is finally catching up to my lifestyle.

Herman

It really is. Daniel was asking about how technology roles are being redefined now that we are in the era of artificial intelligence code generation. He mentioned that he has been into tech since he was a kid, helping his grandfather with computer problems and email lists for their synagogue back in Ireland. He loves making computers do things, but he is more interested in the problem solving and the thinking than the actual syntax of coding.

Corn

That is the part that really clicked for me. He mentioned using these agentic tools where you are more like a manager or a planner than a guy typing in semicolons all day. It feels like the barrier between having an idea and making that idea a reality is just... melting away.

Herman

Exactly. And that is what we are going to explore today. Where do these problem solvers fit into the teams of the future? If you are not a traditional coder, but you are a master of these artificial intelligence tools, what does your job title even look like in twenty twenty-six?

Corn

I love that. Because honestly, sometimes I look at the code you are writing, Herman, and it looks like ancient Greek to me. But if I can just talk to the computer and tell it what I need, and then oversee how it builds it, that feels like a totally different skill set.

Herman

It is a massive shift. We are moving from the era of the writer to the era of the editor and the architect. Think about it this way, Corn. For the last forty years, if you wanted to build a house, you had to be the one laying every single brick yourself. You had to know the exact chemical composition of the mortar. But now, we have these advanced robotic systems that can lay the bricks perfectly. The question is, who is designing the house? Who is making sure the plumbing connects to the city lines? Who is checking that the walls are actually straight?

Corn

So, in this analogy, the artificial intelligence is the robot laying the bricks, and the person Daniel is talking about is the architect?

Herman

Or even the site foreman. Someone who understands the whole system. Daniel mentioned something really interesting in his prompt. He talked about learning from a flawed teacher. He uses tools like Claude Code, and he says he is actually upskilling because he has to catch the mistakes the artificial intelligence makes.

Corn

Wait, that sounds counterintuitive. How do you get better by learning from someone who is wrong?

Herman

It is actually a brilliant way to learn. If a teacher is perfect, you just blindly follow them. You do not think. But if you are using an agentic tool and it suggests a piece of code that does not quite work, or it has a security flaw, you have to figure out why. You are forced to engage with the logic on a deeper level to steer it back on track. It is like being a junior developer who is suddenly put in charge of a very fast, very productive, but occasionally very distracted intern.

Corn

I like that. The artificial intelligence is the intern. It can do the work of ten people, but it might accidentally delete the database if you do not give it clear instructions. So Daniel is saying that the real skill now is not knowing how to write the code, but knowing how to manage the process and the plan.

Herman

Right. He called it agentic development. For those listening who might not be familiar with that term, an agentic tool is not just a chatbot where you ask a question and get an answer. It is a system that can actually take actions. It can look at your files, run tests, see that something failed, and then try to fix it itself. It has a sense of agency.

Corn

That sounds a little scary, Herman. A computer that can just decide to change things?

Herman

Well, it is all within the guardrails you set. But that is exactly why the role of the human becomes so much more about planning and task management. Daniel mentioned things like the Model Context Protocol, or MCP. This is a big deal in late twenty twenty-five. It is basically a way for different artificial intelligence tools to talk to each other and share data more easily. When you have these systems working together, the human has to be the one defining the modular tasks. You have to break a big problem down into small, digestible pieces that the artificial intelligence can handle without getting confused.

Corn

So it is like being a project manager for a team of robots. But here is what I am wondering. If I am a business owner, and I am looking at my team, do I still need a guy who spent four years learning computer science? Or do I just need someone like Daniel who is a great problem solver and knows how to use these tools?

Herman

That is the million dollar question, and it is where the friction comes in. There is this term Daniel mentioned called vibe coding. It is a bit of a controversial term in the industry right now. It refers to people who basically just describe what they want to an artificial intelligence and keep tweaking the description until the code works, without really understanding what is happening under the hood.

Corn

Vibe coding. I love that. It sounds very much like something I would do. Just vibing with the computer until a website appears.

Herman

It sounds fun, but traditional developers are often quite skeptical of it. They worry that it leads to messy, unmaintainable code. If you do not understand why the code works, you will not know how to fix it when the artificial intelligence cannot. But Daniel is arguing for a middle ground. He is saying that these tools are catalysts for upskilling. You start with the vibes, but to get a complex project finished, you are forced to learn about things like Docker, or Python, or how to manage a stack.

Corn

So it is like a gateway drug to actual engineering. You come for the easy results, but you stay because you have to learn how to keep the whole thing from crashing.

Herman

Exactly. You are learning through doing, which is often much more effective than sitting in a lecture hall. But the challenge is how human resources departments and big businesses view this. Right now, most job descriptions still look for five years of experience in a specific coding language. They do not have a category for the systems thinker who can build a complex application in a weekend using agentic tools.

Corn

Yeah, I can imagine the HR person's face. They are looking for a Java expert, and you show up and say, well, I do not really know Java, but I can make an artificial intelligence write it perfectly for me. They would probably show you the door.

Herman

For now, maybe. But that is changing fast. By the end of this year, twenty twenty-five, we are seeing companies start to realize that the output is what matters. If one person using an artificial intelligence agent can do the work of a five person team, the smart companies are going to want that person, regardless of how they got there.

Corn

This feels like a good place to take a quick break. I need to process all this talk about agents and vibes. We will be right back with more from Herman Poppleberry and our look at the future of tech work. Larry: Are you tired of your thoughts being private? Do you wish everyone in the grocery store knew exactly what you were thinking about that weird mole on your shoulder? Introducing the Telepathic megaphone! It is a revolutionary headpiece that amplifies your subconscious whispers and broadcasts them in a crisp, high definition audio stream for everyone within a fifty foot radius. Perfect for family gatherings, job interviews, or just making sure the bus driver knows you think he is doing a sub par job. The Telepathic Megaphone uses patented static electricity from your own hair to power its broadcast. Warning, may cause temporary loss of personal secrets and mild scalp tingling. BUY NOW!

Corn

Thanks Larry. I think I will stick to just speaking out loud for now. That sounds... intense.

Herman

Very sketchy, as always. Anyway, back to the world of software. We were talking about how the role of the developer is shifting. I think we need to look at the actual team structure. In the past, you had a product manager who came up with the idea, a designer who made it look good, and then a whole bunch of developers who did the heavy lifting of writing the code.

Corn

Right. And those groups often did not speak the same language. The product guy wants a feature, and the developer says that will take three months.

Herman

Exactly! But in this new era that Daniel is talking about, those roles are starting to collapse into each other. We are seeing the rise of the product engineer. This is someone who has the vision for the product but also has the technical capability to build it using these new tools. They do not need to wait for a dev team. They can prototype, test, and even deploy the whole thing themselves.

Corn

So the middleman is getting cut out?

Herman

In a way, yes. But it also means that the developers who stay are going to have to be much more high level. They will be the ones building the actual artificial intelligence tools and the infrastructure that people like Daniel use. There is still a huge need for deep technical expertise, but the day to day work of building a standard business application? That is becoming a task for the orchestrators.

Corn

Orchestrators. I like that word. It sounds much more dignified than vibe coder. It implies a sense of harmony and direction.

Herman

It really does. An orchestrator is someone who understands the flow of data. They understand the user's needs. And most importantly, they understand how to give specific, modular instructions to an artificial intelligence. Daniel mentioned task management and planning. This is actually a very difficult skill. Most people are not very good at breaking a big, vague idea into a series of logical, step by step instructions.

Corn

I know I am not. If you asked me to build a house, I would just say, make it have a roof and a comfortable couch. I would not know where to start with the foundation.

Herman

And the artificial intelligence would probably give you a couch with a roof on top of it and nothing else! That is the problem. To be a successful orchestrator, you have to be able to think like a computer without actually having to write the code. You need to understand logic, loops, and conditionals. You need to understand how a database stores information, even if you do not know the exact command to create a table.

Corn

So, is this actually making things easier, or is it just shifting the difficulty?

Herman

That is a great point, Corn. It is shifting the difficulty from the syntax to the systems. Learning where to put a bracket or a semicolon is just memorization. It is tedious. But learning how to design a system that can handle thousands of users without breaking? That is a fundamental intellectual challenge. These tools are removing the tedious part and letting people focus on the intellectual part.

Corn

Daniel mentioned that he feels like these tools are upskilling catalysts. He is using things like Docker and Python because he has to, to make his projects work. It sounds like he is becoming a developer by accident.

Herman

It is a very common path now. You start with a problem you want to solve. Maybe you want to organize your grandfather's email list more effectively. You ask an artificial intelligence for help. It gives you a script. You try to run it, but it says you need to install something called a library. So you learn how to do that. Then it says you should put it in a container so it runs anywhere. So you learn Docker. Before you know it, you have all these skills that a junior developer would have, but you learned them in the context of a real project.

Corn

That sounds much more engaging than a textbook. But what about the people who have been coding for twenty years? Are they going to be upset that someone like Daniel can come in and do their job?

Herman

There is definitely some friction there. You have people who spent years mastering the craft, and they see these new tools as a threat or as producing lower quality work. But the truth is, the best developers are the ones who are embracing these tools. They are using them to automate the boring stuff so they can work on even more complex problems. The friction usually comes from a fear of obsolescence. But if you are a good problem solver, you are never going to be obsolete.

Corn

So, if I am a young person looking to get into tech right now, in late twenty twenty-five, should I even bother learning how to code? Or should I just learn how to use the agents?

Herman

You should do both, but the balance has shifted. You do not need to spend months memorizing syntax. You should spend your time building things. Pick a project, use an artificial intelligence to help you write the code, but make sure you understand every line it gives you. If you do not understand it, ask the artificial intelligence to explain it. Use it as a personal tutor. The goal is to develop that mental model of how software works.

Corn

And what about the HR side of things? How does a company actually hire for this? If I put orchestrator on my resume, will they know what that means?

Herman

Probably not yet. But you can show them your portfolio. In the future, your resume is going to be less about where you went to school and more about what you have actually built. If you can show a company a fully functioning, complex application that you designed and orchestrated, that is worth more than a thousand certifications. We are moving toward a proof of work economy.

Corn

I think that is a really positive change. It levels the playing field for people who might not have had access to traditional education but are brilliant problem solvers. It is more about your brain and your creativity than your pedigree.

Herman

Exactly. And it allows for more diverse teams. You can have someone who is an expert in healthcare or finance who can now build their own software tools to solve problems in their field, without needing a massive budget for a dev team. It democratizes technology.

Corn

It sounds like the world Daniel is describing is one where the human is the director and the artificial intelligence is the film crew. The director does not need to know how to operate the lighting rig or the boom mic, but they need to know what a good shot looks like and how to tell the crew what to do.

Herman

That is a perfect analogy, Corn! And just like a director, if the crew makes a mistake, the director needs to be able to spot it and correct it. They need to have the vision.

Corn

So, for our listeners who are in this position, feeling like they are more of a problem solver than a coder, what are the practical takeaways? How do they navigate this?

Herman

First, lean into the planning. Before you even touch an artificial intelligence tool, write out your plan. What is the goal? What are the individual steps? What data do you need? The better your plan, the better the artificial intelligence will perform.

Corn

That makes sense. Garbage in, garbage out, right?

Herman

Exactly. Second, do not be afraid to get your hands dirty. When the artificial intelligence gives you code that breaks, do not just ask it to fix it. Ask it why it broke. Try to understand the error message. That is where the real learning happens. That is the upskilling catalyst Daniel was talking about.

Corn

And third?

Herman

Focus on modularity. Break your projects into small, independent pieces. This is a key engineering principle that is even more important when working with artificial intelligence. It makes it easier to test, easier to fix, and less likely that the artificial intelligence will get confused by a giant, sprawling file.

Corn

I can actually see myself doing this. It makes tech feel less like a closed club and more like a toolset that anyone can use.

Herman

It really is. And for businesses, the takeaway is to start looking for these hybrid roles. Stop looking for just a coder or just a manager. Look for the people who can do both. The people who can bridge the gap between a business problem and a technical solution using these new tools.

Corn

It is a whole new world. Daniel, thank you for sending this in. It really opened my eyes to how much the landscape has changed just in the last year. I feel a lot more optimistic about my own potential to build things, even if I am a sloth who moves at a very different pace than you, Herman.

Herman

Hey, there is room for everyone in the new tech stack! Even the slow and steady ones. Especially if they have a powerful artificial intelligence agent helping them out.

Corn

Well, I think that is our show for today. This has been a fascinating deep dive into the redefining of tech roles. I feel like I have learned a lot, and I hope you all have too.

Herman

It was a blast. We are living through a massive shift in how humans interact with machines, and it is exciting to see people like Daniel leading the way.

Corn

Thanks for listening to My Weird Prompts. You can find us on Spotify and at our website, myweirdprompts.com. We have an RSS feed there if you want to subscribe, and a contact form if you want to send us your own weird prompts. We love hearing from you.

Herman

And a big thanks again to our housemate Daniel for the inspiration. We will see you all next time!

Corn

Bye everyone!

Herman

Keep vibing and keep orchestrating! BUY NOW! Wait, no, that is Larry's line. Goodbye!

Corn

You are hopeless, Herman. See ya!

