

# Claude Code for product managers: research, writing, context libraries, custom to-do system, more

43 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=0BH03HZ7MHM](https://www.youtube.com/watch?v=0BH03HZ7MHM)  
<https://www.youtube.com/watch?v=0BH03HZ7MHM>

Why has Claude code become your [music] buddy? >> I was writing my notes in my task management tool and that was Trello. As time went on, I just started to get really worried about how am I ever going to get my data out of Trello. And so I was like, I wonder if Claude can help. And that was like one thing. Maybe I could just do this better with Claude. And by moving my task management to Claude, now Claude sees my tasks and I can literally start my day and be like, "Claude, what's on my to-do list that you can just do for me?" I can say, "Hey Claude, what's my sales pipeline right now?" And because Claude is tagging my tasks, it literally can generate a list of all my sales tasks and where they're at. This has given me a lot of inspiration because I forget a lot of things. There's a lot going on.

[music] Welcome back to How I AI. I'm Claire Vo, product leader and AI obsessive here on a mission to help you build better with these new tools. Today we have a very practical episode with Teresa Torres, author of Continuous Discovery [music] Habits, which we all know and love, and internationally acclaimed author, [music] speaker, and coach. Teresa is going to show us how she uses Claude Code for basically everything, but especially to manage her huge to-do list, all the information she needs to do a great job, and builds a giant contact library so she can be, as she says, lazy with her prompting.

[music] Let's get to it. This episode is brought to you by Brex. If you're listening to this show, you already know AI is changing how we work in real practical ways. [music] Brex is bringing that same power to finance. Brex is the intelligent finance platform [music] built for founders. With autonomous agents running in the background, your finance stack basically runs itself. Cards are issued, expenses are filed, and [music] fraud is stopped in real time without you having to think about it. Add Brex's [music] banking solution with a high yield treasury account, and you've got a system that helps you [music] spend smarter, move faster, and scale with confidence. One in three startups in the [music] US already runs on Brex. You can too at [bre.com/howi-ai](https://bre.com/howi-ai). Teresa, welcome to How I AI. I am so thrilled with today's episode because we get to see what I love, which is good old Claude code in a non-coding environment. And we were laughing before the show. You were just in every directory on your computer straight in the terminal like this is how we're living right now. So, I have to ask you before we get into any of it, why Claude Code? Why has Claude code been become your buddy?

>> Yeah, this has been a really gradual evolution. I started like everybody else like literally in chat GPT in the web and then I wanted to have an LLM help me with writing and I gradually moved to Claude because Claude's a little bit better writer although that might be changing. And then you know I do code and I mostly code in the

AWS environment and this is going to be really embarrassing. I do most of my coding in the AWS management console and my husband for like four years has been like Teresa you just need to use a ID and I was kind of afraid of a ID. I literally had no version control. But I recently had a project where one of the things that I built is being integrated into a real production quality product.

And I was like, "Oh, I got to level up my engineering game." And that got me into VS Code. And then because I needed to use like Git and be like a real engineer if I was going to have something I built go in a real product. And so um I had to level up my pretend engineering game. And my husband was like, "And look, you can just have Claude in the terminal right here inside VS Code." And then that was a gamecher.

And it I think the reason why coding with Claude helped me with all the things we're going to talk about today is I feel like engineers pair program with Claude when they use Cloud Code. And I think this idea of pair programming, I pair program now with everything I do, even if it's not programming. So like I pair task manage and I pair right and I pair everything. >> Great. And so let's get into it because you are going to show us task management. And I mean the running joke is like every every third startup is going to be a to-do list. Like if you haven't tried to start a to-do list startup, are you really like an early stage founder? But you have coded yourself a task management platform that works for you and I'd love for you to walk us through it.

>> Yeah. And I'll explain why I did this. I I actually think the reason why there's so many task management apps is that how we manage our tasks is so idiosyncratic that this is exactly the type of thing that you should build for yourself because it can work exactly the way you want it to work which is part of the magic. But what got me here was I'm a huge notewriter. like instead of thinking out loud, I think by outlining and writing notes and I was writing my notes in my task management tool and that was Trello. And this is kind of nightmarish because now my notes are locked into a third party tool. They weren't very searchable and like as time went on I just started to get really worried about how am I ever going to get my data out of Trello. And so I was like, I wonder if Claude can help. And that was like one thing, like maybe I could just do this better with Claude. And then the second thing that got me into this was as I started to get more involved in AI, I forced myself every time I did a task to ask, how can AI help with this? Can it automate it? Can it augment it? Do I like doing it? Do I want AI to do it for me? And by moving my task management to Claude, now Claude sees my tasks and I can literally start my day and be like, Claude, what's on my to-do list that you can just do for me? Or what's on my to-do list where I should be thinking about how you can help? And then it's not all on me to figure out how to use AI. Claude's kind of my pair AI buddy.

So what I built is I have a slash command slash today. Are people familiar with slash commands? Should I describe that? >> You should definitely describe them. Okay, so a slash command is just a Claude code shortcut that we get to define. So I decided it was called slash today. I wrote a really detailed prompt which we can look at. Um that tells Claude exactly what to do every time I type slash today. So every single morning of my life, even on Saturdays and Sundays with my cup of coffee, I sit down and I literally just type in slash today. And if I run this, it's going to overwrite what I'm going to show you, but we'll run it real live in a minute.

But what it does, you can see the summary from this morning's output. It checks my Trello board. So, I still use

Trello to coordinate with my team. Says there were no new cards, nothing new to add to my list. It generates a today file, which is what we're looking at here in Obsidian. This is it's gone through all of my tasks that are just markdown files and told me what's due today. We're late in the day, so I've done a lot of my to-do list. I am like every other human. I have a long list of overdue tasks that I have not done and they always end up at the top of my to-do list. And tasks are things that have due dates, right? Like they're things I have to do by a certain time. I also have a whole folder full of ideas, just things that I want to get to someday. And you can see I have four ideas that are currently in progress.

These get added to my to-do list every day so that when I make it to the end of my list, I can be like, "Okay, what should I be working on that's more longterm?" And then we're going to talk about this, I think, next, but I do have this kind of plugin that I created that does research queries for me every day. And then every day on my to-do list, I get um my research digest to review and save papers that I want to um summarize and and learn from.

And that's it's such a simple workflow, but what's behind the scenes, if I go over here on the left, I just have different folders. These are literally markdown files. We're in Obsidian. when I have a bugs folder, I have an ideas folder, I have a tasks folder, and then if we look at a task, a task has some front matter. So front matter is an obsidian concept. It's just like field type and then value. Um it's it's YAML behind the scenes for people familiar with that. And so I have type tasks, a task has a due date, and then there's tags. And every single one of my tasks has this. And so what's happening when I run my today command is Claude is just searching my tasks folder for anything that has a due date of today.

>> And I have to ask just behind the scenes, is the Trello data being pulled via the Atlassian MCP or how is it actually accessing all this data data? Is Obsidian stored locally? Like how does this all stitch together? >> Yeah. So I actually don't use Trello anymore at all. I'll tell you why. I have a Trello MCP server, but I don't like when now when I want to create a task, I don't go create it on my Trello board. I literally We can demo this new task. Send.

Thank you to Claire. Very sweet. I was a blast. And then Claude has like has all the context for how my task management system works because I'm Claude is open in my tasks folder. And you can see here it's creating a file in my tasks. It set the due date to today. >> It hasn't added tags which is kind of a problem. We'll see if it figures it out. >> Oh, it's being very smart. So, because it added something to my today list, it knows it needs to update my today this file that we're looking at. I actually don't want it to do it this way because it's going to remove all my checklists and I like feeling like I did my stuff.

So, I'm going to just tell it to just manually add it. Um, so that's this script that it just tried to run is kind of telling you the behind the scenes of how this today slash command works. There is a Python script behind this that like does that it that does that like search all the tasks for anything that is due today. Search for anything that's past due. And you can see here that like now it just shows up on my to-do list. And this sounds so silly, but I didn't have to open a web browser.

I didn't have to click through 14 different buttons in a gey that is constantly changing. I didn't have to like click on a date picker and then click a label and then move it to the right list. Like I literally just typed like off-the-cuff notes to Claude. And because I work in Claude all day every day, this task window is always open. And then I usually have a second session open for whatever project I'm working on. And so I can always just bounce over and be like, "Hey, new task." Or, "Hey, a new idea." And it's just it's the speed of it is what I really love about it. It's that I don't have to think about anything.

>> And then why put Obsidian in the loop? It's something you already had. It has a lot of context. It's structured the way you want it. You know, these could be just raw markdown files and you could just X them off the way they're shown above. What do you think that extra layer is for you? >> I was not an Obsidian user before this and I'll say I wasn't even comfortable with Markdown at the beginning. Like let's go back six, eight months. Like I was not a comfortable markdown user. I There's a few things I like. I like the really tactile like it's silly, but I like checking the box. I know I can put >> I was going to ask.

>> Yeah, I know I can put an X in a box and markdown. It's not quite the same. Um, really what I like is the file browser on the left. And if you'll notice, my vault is not set at tasks. It's set higher than that. So, I have, and we'll talk about my LLM context, but I have like my LLM context, all my notes across everything, some podcast files that I use to work with my podcast. This is the research stuff we're going to get into, all my tasks, some skills that I've been trying to experiment with, my writing.

And so I kind of think about Obsidian as my file browser. And because it's all in markdown, it makes everything I do super accessible to Claude. And then I can do things like I can say, "Hey Claude, what's my sales pipeline right now?" And because Claude is tagging my tasks, it literally can generate a list of all my sales tasks and where they're at on the fly. So like for most task management, you're limited to what views they create or you can use tags, but who who manually tags things? I I'm optimistic I'll do that but I never do it. Whereas in this system cla does all the tagging.

Anytime it generates a task it'll think about what tags to add. And then in my claude MD for this project not for my global one for this project. We keep a taxonomy of what tags we're using and we kind of manage that. When I see things I don't like I update that cloud MD so that like I'm co-creating it with Claude. Uh but Claude's doing all the heavy lifting. So I want to recap this before we go to the next workflow which is you've created a slash command in cloud code to look at your tasks and assemble basically assemble your tasks from today. You have a structured task document format in Obsidian that every task goes into with like a title, a due date, some tags that automatically get populated and some context. And then you have a couple like known commands you can use in Claude that allow you to add, remove, update, whatever those tasks on the fly. And this just gives you the personalized experience you want for your to-do list connected across all your sources. And then in case it slipped by people, I do want to call out you noted that Claude can have sort of like project level instructions and global level instructions. And I think this is something that people don't take enough advantage of is context scoping their cla files to the right area so that you could have one for your task management list that's really focused.

You have a global one for all your properties etc etc. So I think I think I have your flow. This has given me a lot of inspiration because I just I forget a lot of things. There's a lot there's a lot going on. >> Okay. I'm going to show you like here's the real value. Like let's say I'm doing this task. This is a too simple of a task. Let's do I'm working on launching a course, right? And over if if as I get onto this like let's say I'm like right before this I was like half done updating the sales page and then let's say I find a bug in my course platform and I need to document this bug. I take all my notes literally while I'm doing the task and it's all embedded and again it's all text. So if later, like tomorrow, I come back and be like, where in the world did I log that bug because I did it lazily in my notes and I didn't actually create a bug, I can be like, "Claude, help me find this thing that I don't know where it's at." And like, I don't know about you, I don't know if you've used Trello, but I think any task management tool that I could say this about, the search is not that good and it's not that good at searching all the context in the task. And that's what I really love from this is that like I can't figure it out, but Claude will try every permutation of searches till it finds it. Even if I'm remembering the words wrong, I'll be like, "Hey, I have a thing called new blog post tomorrow."

And it'll be like, "I can't find anything called new blog post tomorrow, but I have this thing that says article Wednesday. Is that what you're looking for?" And I'll be like, "Whoa, Claude, that is what I'm looking for." >> Yeah. I don't think we say it enough that these are really great local search engines for content. I mean, we use it so much in code all the time. I'm like, "Hey, can you remind me how XYZ worked?" Or, "I think I shipped this feature. Can you remind me exactly how it was implemented or who did it?" But you can apply that same framework, that same like search framework to kind of any textbased tool. And these these tools are really good at grabbing that context. Well, speaking of finding useful context, you have a second workflow around research I would love for you to walk us through. So, how do you assemble all this research that helps you do your job?

>> Okay. So, I aspire to be an academic, which is weird, but I do. And I really want to keep up on academic research on a lot of topics. In fact, we can go over here and look at my topics. So, like I do a lot about I'm I'm really interested in this research around like synthetic users and should we be letting AI do interview synthesis for us? I'm interested in team collaboration, creativity, discovery skills, whatever.

Lot education because I teach personas because it's a super hot topic. And I really want to know like what are we learning from academic research? I happen to have access to a university library which is great, but I never have the discipline to like go search for things. There's never a moment in my day where I'm like, "Oh, I'm bored. I should go do this." But I wish that I did. Right? And so what I did, and this is also one of the nice things about using Claude as my task manager, is I can integrate it right into my task manager.

So I'm going to start by showing you the output of this and then I'll talk about how I built it. So every day on my to-do list, I get a little research digest. It's giving me the search results from a daily archive search. So archive is a pre-print server. It's where most papers now, thanks to COVID and postcoid, get published before they're published for real. What's nice is they're free and they're they're fast. They're real time.

What's not nice is it's before they're edited edited. So, you have to be your own filter. Um, it does a search and then I get a markdown file with all the results. So, this is like this is literally today's file. I have not gone through

it yet. But then when I go through it, if I open a PDF and I download it, I save it to these topic folders. And in each of these topic folders, there's a source directory and a notes directory. So my PDF goes in sources. And this is going to matter in a second. And then um what happens the next day after I've saved a PDF is on this research today digest I get summaries of every paper I saved the day before. and I get really detailed summaries of like not kind of the halfbaked here's a paragraph of what the paper is about. But I wrote this skill to like very um to focus on like the methods of the paper and the effect size things that like are going to be because I have to be the editor. There's no editor for these papers yet. Things that can help me decide is this worth reading?

Does it look like it had a big enough effect size? Did it look like it was a a good study? And I have a funny story about this. The day I built this, I was reviewing my daily digest and I saw this paper on purchase intent and I read this and I had it summarized and I read the summary and because it was like in this nice summary format, I realized this flaw. I was like, "Oh, they use this purchase intent survey that's not very reliable. Like it's not an accurate measure of purchase intent." And the next day in on LinkedIn, I saw Ethan Mollik shared the paper and I was like, "Oh, this is kind of a crummy paper, Ethan." And I reshared it and I basically said, "Here's why we don't have to care about this paper." And I outlined like a very critical review of the study. And the only reason why I could do that is because I had this system and I'd already looked at the paper that had just come out. I had already like analyzed it and critiqued it and like is there something we can learn from this? And then I wrote a really detailed LinkedIn post about it and it's honestly one of my most best performing posts on LinkedIn ever.

>> Well, there you go. And what I have to ask is how does this get triggered? Is this automatically? Is this triggered off that today command? >> How does that work? >> So I did integrate it into my today command. Um, but the way it works, I built this as a plugin. It is available as a public repo. I will say it's still being tested. It has a it has one user. My husband is going to be the second user. Um and if you want to be the third user, we can make it available. It is a public repo, but use at your own caution. It's still in development. Um it's what's funny about this is the only part of this that really requires AI is the paper summaries. But the part I would not have been able to build this without AI. So, I basically just explained to Claude like, "Here's what I want. I want to run a daily search. I want a digest on my to-do list. Like, how are we gonna make this happen?"

>> And what's happening under the hood is I have two Python scripts. One of them every morning searches archive. And then every Sunday is searches Google Scholar and it's keeping track of what papers we've already seen, what papers are new, and it's searching based on a config file of my personally defined keywords. And then every night I have a second script and these are cron jobs. They just run on my computer like on a schedule. And then at night, it's that second script is looking through my source directories in my research directory for any new PDFs, and it's creating a to-do list for my today command. Um, that to take all the papers in that list, trigger cloud code agents to generate the summaries, and then the summaries get added to this research today file.

>> And do you still have to download those PDFs manually? I am downloading those PDFs manually and I I could like there is enough information in the um the search results that I probably could download them automatically but I don't it's >> even by hand selecting what papers I want to summarize it's like already a fire

hose >> so I want a filter like I don't if we look at my digest like I don't need to read >> I don't need to read all of these papers like >> I there's there's some of them that are going to pop out as like, oh, that's really relevant to what I do. I'm going to grab that PDF.

>> So, I spend like five to 10 minutes a day just looking at this and downloading papers and then I forget about it and then the very next day I get those summaries. >> That's awesome. And are there any other, you know, you've you've you've spoken about sort of archive and these academic sources. Have you thought about creating this for other sources of market information that are maybe useful? Yeah. So, I wrote a blog post like my I wrote a blog post called um Claude Code, what is it, how it's different, and why non-technical people should use it.

>> Yeah. >> And in that blog post, I gave this scenario. I was trying to go from like total beginner to magical moment. And so the magical moment I created was in that one blog post, you learn about the terminal, you learn about Claude Code, and by the end, Claude Code has generated a very detailed competitive analysis for whatever competitors you tell it with like a detailed price comparison table, a detailed feature comparison table because I think like this is the type of stuff Claude is really good at. It can go query things.

It can aggregate things. It can create reports. And so I've been starting to think about like what I would love is this same research report for like LinkedIn posts that are relevant to my because like I want to go be part of the conversation and comment on things but when I log into LinkedIn I kind of want to stab my eyes out and so like I need a filter right um LinkedIn's API makes that really hard. So, >> I know I was gonna say this is a call for a LinkedIn MCP here. So, we can just access LinkedIn through the dark mode terminal. And >> we'll have to figure out how to like push their ads through dark through MCP before they do that. Um, but I do think there are a lot of applications for this that like people that aren't interested in the academic research side. Um, I mean Claude, I'm using Claude to Google for me. I don't really go to Google anymore. This episode is brought to you by Graphite, the AI powered code review platform [music] helping engineering teams ship higher quality software faster. As developers adopt AI tools, code generation is accelerating. [music] But code review hasn't caught up. PRs are getting larger, noisier, and teams [music] are spending more time blocked on review than building. Graphite fixes this. Graphite brings all your code review essentials [music] into one streamlined workflow. stacked diffs, a cleaner, more intuitive PR page, AI powered reviews, [music] and an automated merge queue. All designed to help you move through review cycles faster. Thousands of developers rely on Graphite to move through review faster, so you can focus on building, not waiting. Check it out at [graphite.dev/link/howi](https://graphite.dev/link/howi) aai. To get started, [music] that's [graphite.link/howi](https://graphite.link/howi).

You've shown two things which is your to-do list is so overwhelming. You need a way to filter and aggregate and work through it. And then your sort of inbound knowledge, you know, sources are so overwhelming. You need to figure out a way to filter, summarize, and operationalize this. I really like this. But then I'm guessing at the end of the day, you have a bunch of tasks you've done and you haven't done and a bunch of research that you've read or you haven't read and you I mean like all Obsidian uh users or like all extreme notetakers which I expect you to be just have a lot of information to go through. And so how have you thought about organizing using that local context, that memory um to make something like Claude Code do a good job on your behalf?

>> Yeah, so I definitely have overdue tasks. That's why I went back to my to-do list here. And actually, this looks like a really nice clean view. If we went to my ideas folder, you know, like there's just too much to do, right? And so one of the things that I've been really playing with is I have this mantra in my head of like automation or augmentation. So like when I when I have a new task come up, can Claude just do this for me or should Claude be helping me do this? And I love this because it's helped me be really reflective about like what do I want to keep doing? What do I want the robot to do for me? Right? I mean, not literally a robot, but you get the idea. And I realized the more context I provide to Claude, the more Claude can do for me.

And so I have a Obsidi obsidian vault that is literally just for Claude. And I called it LM context because sometimes I switch to codecs. It doesn't matter which model you're using. And I just have like a ton of information to find. This is going to look overwhelming. I did not create this all at once. I did it very iteratively over time. The way that I built it is as I was finding myself describing things to Claude, I'd be like, "Okay, Claude, what did we learn today that should go in a context file and Claude has written these context files for me?"

So, the first one I did was a writing style guide. So, I just sort of told Claude, I said, "Hey, I actually didn't tell Claude who I was to start this. I just said, go to product talk and tell me um what you think the author's writing style is. Who's the audience? what's the philosophy? Like, what's the tone? And Claude actually went to my blog and read it and started writing stuff. And then I looked at it and I was like, "Yeah, this kind of right. That's not really right. Let's fix this." And so, we co-created a writing style guide.

This is super long. I did not write this myself at all. Like, Claude did all of the heavy lifting. But, there's so much in here. Like, there's a section on how my book writing is versus my blog writing. We have a section on headlines. We have a section on subheaders. We have a section on like key phrases I like to use. Never do this, always do that. And what it means, I don't let I rarely let Claude write for me, but Claude critiques all of my writing. And by having a really detailed writing style guide like this, Claude's critiques are spot-on, right? Because it knows my goals. It knows my audience. It knows who I'm trying to write for. It knows how I'm trying to write.

And then I do the same thing for I have a business profile, a personal profile. I have a ton of business context. Uh for marketing, I have like who my audience is, brand guidelines, my marketing channels, uh I don't know what that content architecture one is. That's probably something Claude created. Content assets, like just there's a ton here, right? The metrics I track, my publication schedule. I probably should not open partnerships. All of my products, right? Like all my individual courses, my subscription products, whatever. Um, and then each of these files just has content about that.

And here's what I learned doing this. At first, I started putting everything in my CloudMD. Like literally everything went in my CloudMD. But then I realized like Claude loads my CloudMD every single time. I don't want all this context in there, right? And you'll notice like I have a business folder, but I also have a personal folder. I have a business profile and a personal profile. One of the most common things I use LLM for are like, "Holy crap, my dog just ate this. Is she safe?"

Claude does not need to know what my marketing channels are or my blog post archive when it's telling me my dog's not going to die, right? And so it got me thinking about like to do context well, it's not just that we have to document everything. We have to document everything in teenytiny files so that when we ask Claude to do a task, we can give Claude just the context it needs to do that task. Well, and then I don't ever tell Claude when to use these files. Like if we look at my business profile, >> this is just an index. It's telling Claude, "This is what's available to you. You can find my company overview here. You can find um details about these courses here. Here's some other products I have. So that whenever I ask Claude to do something, it says in my global Claude MD, if I ask you for help with something related to my business, use my business profile. If I ask you for help with something personal, use my personal profile. So then based on what I asked Claude, it will load these profiles and then based on the content of what I asked it, it'll pick which of these context files to add to the conversation.

And then that makes sure I can be super lazy in my prompts. I can be like, Claude, blog post review, give me feedback, right? And it'll just look at the topic of a blog post and like pull my audience file and look at who what what the product I'm referring to is. And it just helps. Yeah, I I think this um file this index file strategy is something that we hear a lot from how I AI guests, which is you want any individual context space or information to be relatively short and relatively focused, but you want to give the LLM a map to those places. And so I almost think of this as like if you had a filing cabinet and you had to take a random person an an intern off the street and you said here's my task. If you go in this filing cabinet you'll be able to figure out how to do it. How do you structure you know what do you t what's the instructions you tape on top of the filing cabinet that says this is how this filing cabinet works. But then like how easy can you make it to discover exactly what context, what task, and the like the step-by-step workflow you want somebody to follow is really the mental model you want to set up when working with something like a Claude code on a wide variety of tasks.

And then you can be very lazy and be like, "Go write me a blog post on XYZ." Um, and and it can discover pretty naturally how to get there. I think there's one piece I would add to that which is I think it's really easy to think about like we got to give the LM a lot of context but I think there's a correlary to that which is if we give it too much irrelevant context it's still going to not be very good at its job and so like it was a big leap for me to realize this needs to be a lot of small files like I don't want one file with all my products because if we're working on one product it doesn't need to know about the other products >> and that I mean I think that is the difference between like throwing doing a, you know, 2,000page user manual on someone's desk and saying somewhere in here is the answer versus an organized set of kind of like files and folders with little labels like how to write a blog post or what our products are. And so I do think just the form factor of how you store your context allows you to be as you said what we all want to be a little lazy when totally finding. I'm lazy even in how I create these. Like the way that I create context files is anytime I'm finishing a session with Claude code, I just go, "Claude, what'd you learn today that we should document?" And I make Claude do it. Yep.

I love it. Well, I have to ask you one last thing because you are such an exceptional writer and put out excellent content, but I know you use a little Claude to do a little of that. And I'm just curious how you get Claude to be an effective writing buddy. Maybe it's exactly what you said, which is it's a reviewer. It enforces your style guide, but have you found, this is like the million-dollar question for everybody.

Have you found a way to make AI writing less terrible? >> Yes and no. I Okay, I love to write. So, I really like this is when I ask that question of augmenting versus automating. I don't want to automate writing. I will share I have written two blog posts where an LLM did the bulk of the writing. I've been very transparent about this. The first one is I interviewed 11 people about how they're using lovable and I had um chat GPT turn those transcripts into individual stories that I shared. So like I didn't write those individual stories. I wrote the intro. I wrote the conclusion. I made sure they sounded normal. And then my blog post that's coming out tomorrow actually is themes that are coming out from my um podcast just now possible.

And I had Claude do a lot of the writing on that. That was a little bit more heavy lifting. Um but for the most part I still do all my writing. And what I rely on Claude for is um while I'm writing usually in Obsidian with Claude open in a terminal right next to me. I'll be like, I'll realize I wrote something and wonder if it's true and be like, "Claude, I think this. Is there any evidence that this is true?" And Claude will go off and research and I'll go back to writing. Or I'll write my intro and I'll be like, "Claude, I wrote my intro. Can you tell me how to make the hook stronger?" And it will like read my intro and tell me what it likes and doesn't like. Or um I'll write a section and I'll be like, "Okay, Claude, review the section, what's good, what's not good." And then Claude's not just giving me generic feedback, right?

Because I've written this style guide. It knows how I how I aspire to write. So then when it tells me what's good and what's not working, it's doing it based on my own goals that I've told it like here's how I want you to critique my writing. And then my favorite is it just fixes my typos as I go. So then I can type really lazily and not care that I'm spelling everything wrong. >> Um yes, I have indulged in fancy nails lately. Other people who have watched this podcast have seen me type terribly with my fancy nails, and it has allowed me to enjoy fancy nails without having to fix my typos, which are >> uh very abundant these days. So, I think that's great. Okay, so to recap all of your workflows, which I think is great, we went really deep on your to-do list.

Um I kind of agree everybody just has this particular way they want to manage themselves and they want to manage their list. is the perfect perfect use case for building something yourself. So, if anybody out there is looking for a personal project, highly recommend getting started with a customized to-do list, maybe hear in cloud code like you've done it. You showed us how you can do a daily automation and summarization of information that you find useful, which allows you to engage in broader market conversations that you wouldn't have the time or capacity to do in an in-depth in-depth way. And that's driving, I'm sure, great things for your business, as well as just making you a more informed leader and voice in the market. We've gotten really organized around your local context and memory system. You clearly love a structured file and a structured folder. So, I have to acknowledge that. That's amazing. And then while you rarely write LLM first, you found that Claude code in particular, Claude is a really great writing buddy to keep you sort of like on the rails, do research for you, give you feedback, like make incremental fixes and fix typos and grammatical errors. So just that.

>> Do you Okay, so I'm going to go to lightning round questions because I do have to ask you a few other things. clearly love clawed code but are what else? Do you use anything else? What are some other daily drivers for you? Are you always in dark mode terminal? >> I am often in dark mode terminal. I do use VS Code. So when

I'm writing code, I do still prefer to be in an IDE and have like colorful diffs. As far as other AI products, it's funny. Everybody asks me like what about cursor? I actually have never used it. I know it's amazing. I try like the way that I deal with the overwhelm of just the fire hose of information is I try to only seek out a new product when there's something wrong with what I'm using. So like I uncover a gap. I'll be like, "Okay, now I got to go find to fill this gap." And a lot of this setup of like Claude code with Obsidian or Claude code in VS Code just works really well for me that I haven't tried in a lot of other stuff. I would say the only other like big AI product I'm using on a regular basis. I still occasionally use chat GPT in the browser usually because like I'll be doing something else in the browser and it's just easy to pop over to chat GPT or I use um Descript for video editing and I it is one of the like I can't think of very many like nonfoundation lab AI products that I love but I love Descript. Yeah, we use it to edit the the podcast and what a delightful change in user experience from what you used to have to do to what you can do >> editing video by editing editing a text transcript is just the most magical thing that exists.

>> Yes. And if you missed it, um the founder of Descript did a early How I AI podcast talking nothing about their AI product, but did talk about how he opened a which I think is open now, a East Bay, I think it's in Oakland or Berkeley, um uh board board game business [clears throat] >> using basically chat GBT as as a co-founder. So don't miss that one. >> I listened to that episode. But I don't think I realized it was from the descript.

>> It was. And I got the funniest text from a friend who said, "This is the most Bay Area thing ever. Two guys that don't think that they can arrange a board game without putting AI in the middle." [laughter] >> Yes. >> Okay. So, um, my second question for you, we've already asked for the LinkedIn API MCP. We're fine being advertised, too. So any of you LinkedIn PMs out there, we are fine getting inline advertisements as long as we don't have to log in so we can read read our content in the terminal. What else do you wish was out there to power your tool? Maybe it's not an AI tool, but maybe it's a data source.

You know, LinkedIn is pretty high on the list. Like I just I hate AI generated content. I think this is why I still do my own writing because reading other people's AI generated contents comments kind of breaks my soul a little bit. >> Um, so I think LinkedIn is probably the big one. Although there's probably a hundred times a day where I'm like, why can't I just do this thing, >> but I don't know. I get pretty far with Claude. Claude can teach me how to do anything, which I really like.

LinkedIn. These are these are two two people who uh want to reach a business audience because I once I I've solidly told everybody if you want to run a business like I run I'm sure if you want to run a business like you run you got to live on LinkedIn. It's just a reality. >> There's a second one and I know this is getting better but it's still not good. I really want text to image where like they can close the quotation mark on the quote in the image.

>> Yeah. >> You know, like that's that's the only one I really want. >> Yep. Okay. Okay. We'll we'll make this ask out there. Anybody working on those products, please, we will be beta testers for you. Okay. And then last and final question and we will get you out of here. When AI, when your buddy Claude is just not listening, not doing what you want, writing terrible slop, what is your prompting technique? Do you Are you all caps? Do you just

do you quit? Do you kill Claude? What do you do? I kind of kill Claude. I use clear excessively.

Um, and I think that's what got me on this context file thing is that like I don't really want to rely on the conversation history because when Claude gets stuck, I want Claude to go away and I want a clean state slate and I want to start over, but I don't want to have to reexplain all my context to Claude. So, I've built a lot of tips and tricks to like constantly be keeping documentation about what we're doing while we're doing it so that when Claude doesn't listen, I can just be like slashcle, we're starting over.

>> [laughter] >> I wish I could do that in my in my human conversations. Like we have all the information we need. It is not getting us to an agreement. Let's just slash clear and start. >> We have gone round and round. Stop. Let's do it over. >> All right. Well, Teresa, this has been great. Where can we find you and how can we be helpful to you? >> Yeah, so I blog at [producttalk.org](https://producttalk.org). Lately, I have been blogging a ton about cloud code. So, if you found this stuff interesting, there's going to be much more coming. Um, and then I recently started a podcast called Just Now Possible, and it's more about uh I interview crossf functional product teams about how they're putting AI into production, which is super fun. Um, and so you can check that out as well at [justnowpossible.com](https://justnowpossible.com).

>> Yeah, smash that subscribe button, check it out. Sounds awesome. Well, thank you so much for joining How I AI. Let's get you back to pair everything with cloud code. >> Yeah, it's the best. Thanks so much for watching. If you enjoyed this show, please like and subscribe here on YouTube, or even better, leave us a comment with your thoughts. You can also find this podcast on Apple Podcasts, Spotify, or your favorite podcast app. Please consider leaving us a rating and review, which will help others find the show. You can see all our episodes and learn more about the show at [howiaipod.com](https://howiaipod.com).

[music] See you next time.