My tech stack - Staying productive in the age of AI

25 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=ME286AXVKUS
https://www.youtube.com/watch?v=Me286aXvkUs

SUMMARY

The speaker discusses significant changes in their developer workflows over the past three months, focusing on the evolution of AI models and their impact on productivity. They highlight the rapid advancements in large language models (LLMs), the shift towards desktop applications for better organization, and the importance of managing AI usage costs effectively.

- *The speaker has been using LLMs extensively for various tasks, noting their speed but also the risk of generating unmanageable outputs.*
- *Recent advancements in AI models, including the emergence of highly capable models like GPT 5.4X and GLM 5.1, are reshaping workflows.*
- *A trend towards desktop applications is noted, as they help manage cognitive load by reducing tab clutter.*
- *The speaker emphasizes the importance of tracking AI usage and costs, spending around $1,000 monthly on AI tools.*
- *They advocate for a model-agnostic approach to development, allowing flexibility in using different AI models based on their strengths.*
- *The speaker discusses the growing size of AI models and the need for efficient management of resources to avoid escalating costs.*
- *They highlight the potential of smaller models to perform competitively with larger ones, emphasizing the importance of quantization and optimization.*
- *The session concludes with a call to resist vendor lock-in and to leverage open-source models for greater control and adaptability in development.*

Speaker 1: Hello everybody. Today I'm going to discuss with you my current developer workflows, how things have changed over the last three months, what tools I'm using and why. I mainly wanted to do this video because I made one about three months ago and everything's changed since then. I mean, that's to be expected. This industry moves extremely quickly. So as a quick intro, I am a software developer.

Speaker 2: I also do a bunch of other

Speaker 1: stuff like Devrel, applied research, whatever I feel like at the time. And I've been using LLMs for the

Speaker 2: last almost three years now to do

Speaker 1: almost all of my work, or at least I do most of it through the models. This has kind of changed a lot recently. You know, they're very fast and they allow you to do way more than you usually can, but the downside there is that they're so fast that you're just drowning in slop constantly. If you don't really think about every single decision that the model is going to make, then you're going to end up with something that you don't understand.

Speaker 1: And once you get there, it's really hard to unbundle your way out and actually go back to like some kind of working product or some kind of working code base. So I wanted to talk about this a bit and just kind of walk through what's happened lately. So for those of you who don't know, we have had so many new models over the last six months that are like highly capable and you can see that they're kind of converging on a similar score range, going from like 49 all the way to 57 on artificial analysis.

Speaker 1: This benchmark is a group of different benchmarks. So it kind of like groups all these different things together and gives an like an IQ score. You can think of it that way. We could see here that. Oh, we could see here that the top three models are all like, they're all number one right now. The GPT 5.4X High, Gemini 3.1 Preview

Speaker 2: and Opus 4.7 Max.

Speaker 1: We could see that the newest open weight model, Kimi K 2.6 is basically leading now all of the open weight models. And this is a huge LLM, as you guys might know. It's 1 trillion parameters. The next best one, in my opinion on the list is the 5.1 GLM 5.1. So these two models since releasing have been incredible. So let's go over to the GLM 5.1 release and I will tell you why this has been interesting as well as this trend that I'm seeing, which is long horizon work.

Speaker 1: So all of these companies are starting to realize that Everybody is spamming 5, 10 different agent sessions at the same time, which is creating a lot of pressure, cognitive pressure for people. So we see now in this industry a shift towards desktop applications. Like for example here we have the let's go find the Codex app. So we could see here like this is the factory AI app and this is the Codex app.

Speaker 2: And let's see if this has been fixed. I don't think it has.

Speaker 1: Yeah, it has anyway. But you can see here that like these companies are moving towards desktop applications because the mental bandwidth it takes to jump between like 20 or 30 tabs is very high. But if you have a desktop app, you can essentially pin your chats and go through them much easier. Easier. So just fix it and it becomes easier to track. Now I'm fixing something over there, so keep that in mind.

Speaker 1: Another thing that's changed recently is the. And I am probably going to dizzy you to. So the latest model was 3.6 the 27B. So the latest models that are coming out as open weight, open source, etc. Are just phenomenal. So this model here, Quin 3.6.27B is about, I don't know, something like 54 gigabytes if you have it

in full precision.

Speaker 1: And you can get it all the way down to 13.5 gigabytes for the weights if you are using a 4 bit quantization. And if we zoom in on the scoring here so you can see that agentic benchmark. This is the 3.627 B. So this could fit on a phone, the right quantization, the right like phone. It could fit on a phone and it is out competing models that are like 10 times its size at the very least.

Speaker 1: And so this gives you an idea of like what's changing right now. AI inference is becoming more expensive. The corporations and companies that are driving the main frontier models are realizing that the next move is not more terminals and like more distraction, but a user interface like a GUI that has great organization layers to it so that you can basically monitor all of your sessions, save them, create threads and have an experience that's more similar to discord, that's going to allow us to better organize our work in general.

Speaker 1: We could also see another trend in these frontier models is that they're growing.

Speaker 2: Go back to the hugging face page.

Speaker 1: So the latest model that I've been working with has been GLM 5.1 or GLM 5. I've used both. So if we go to GLM 5.1, we can see here this is the latest entry, like the Frontier entry from the Zai company. Their previous flagship model was about 360 billion parameters. Now their flagship is 754 billion parameters. So it's doubled in size essentially. And they're not the only ones. Right. There's this Mythos model that has come out with Anthropic that is claimed to be much, much larger.

Speaker 1: The first 5 trillion parameter model that I know of, like officially announced is Ernie. This is another, I think, Chinese company. So we could see that these models are getting larger and there are good small models that are coming out. But people are kind of addicted to the top of the line intelligence and are not willing to compromise for less. This video is going to go all over the place a little bit, but it's going to tie in together in terms of like what I'm using and why.

Speaker 1: So now my stack has centered around a few different things. One is the. One second.

Speaker 2: Let me see if it's all over here.

Speaker 1: Yeah. One is Codex Bar.

Speaker 2: So if we open Codex Bar, we'll see here.

Speaker 1: So you can see that my Weekly usage of AI has gone up exponentially. With Zai, I'm now using 1 billion tokens a month. With Claude, I am using about 6 billion tokens a month. With Codex, I'm using about 5 billion tokens a month. So I'm ranging between 10 to 20 billion a month on average. And that is a lot. But what

am I doing with all this? So if we go over here, we can look at this app.

Speaker 1: So you can see here that I'm doing a lot of different things. For example, right now I'm trying to debug why my codex broke, so I

Speaker 2: reverted the breaking changes.

Speaker 1: Okay. Restart the app

Speaker 2: and use the Agent browser CLI with Electron to try it out. See if you can get all the models that I want in there without breaking other stuff. Yeah, we can put this.

Speaker 1: So one thing that you're going to notice is that I am using again a lot of different models. Right. So we have GPT, Claude, Copilot, Zai, Kimmy, the local models that I'm running, all this stuff is. Yeah, like I'm using all this stuff constantly.

Speaker 2: So we have five mini local unified models. Let's see if this is going to work.

Speaker 1: I don't think it will, actually. It did.

Speaker 2: Okay, that's nice.

Speaker 1: But not all the models are loading. I'm using a bunch of different models. I'm Using Codex Bar to tell myself, okay, how much am I using per month? Because it's very important to track this information. If you're spending. I'm spending around $1,000 a month on AI stuff. If I'm going to spend that much, I need to know what's going on, why, how, so that I can keep track of how many tokens am I getting. Right? Because they're subsidizing all this usage for us, which means that we get like a bunch of AI for practically free.

Speaker 1: But this is changing very quickly. Like the quality of the models is going down, the amount of thinking budget is going down, the cost per token is going up and all this stuff, like it comes together to form like this image that we're not going to have the inference that we currently have constantly. And so we need to work around that by figuring out what is actually working, what people are trending towards, and then how we can integrate it into our own systems.

Speaker 1: The next thing is I'm using something called Viproxy. This is built on CLI Proxy API. Let's go over to the browser and we can talk about that a little bit.

Speaker 2: CLI Proxy API.

Speaker 1: So when you are using all these different subscriptions, you would have to deal with all of the OAuth tokens by yourself. It would be very complicated. But if you set up CLI Proxy API or Vite Proxy, you get this UI that lets you basically authenticate with all of the different providers. So plot code with Codex and then it puts the model into a local server and. And then you use that local server with a key that you provide.

Speaker 1: And now you can use Codex in plot code. Plot code in Codex, all the different models can be used in all of the different harnesses, which is extremely useful if you're doing what I'm doing and you want to not be broke.

Speaker 2: Only four models are showing up

Speaker 1: anyway. So it's going to be really useful if you're doing this kind of thing to have all of your subscriptions in one place, it's going to make it just much easier for you to manage. Then in terms of like the applications, I am gravitating towards the Codex app for many different reasons. But with Codex, like you could see here, unfortunately you really can't see because I kind of screwed it up. But the app itself is just like very well put together.

Speaker 2: Let's see if I can get any of the files.

Speaker 1: The app itself is very well put together. So I could open these projects. Each project has sessions. I could create threads of Those sessions, there's a browser in the app. It is relatively cheap compared to all the other options on the market.

Speaker 2: Right now.

Speaker 1: The next best thing from a desktop application perspective is the cursor app, but it is expensive because you have to pay API prices. And I'm not adding on to like all of the stuff that I already pay for. I don't want to add more things to it. And when we take a look at like this application, this has been helping me keep better track of meta work. So meta work is like what projects am I working on?

Speaker 1: What are the progress of all these projects? How am I taking a solution I find in one place and then applying it somewhere else? All this stuff is pretty new because generally if you were a contractor, let's say, and you are working on multiple different projects over the years, all of the stacks tend to be different, all of the different UIs are styled completely differently. But realistically what we could do is have a quality core thing that we use everywhere, that we reuse these UI components, we reuse the logic, we reuse the convex data, the skills, and all this other stuff that typically would take a very long time to implement from scratch.

Speaker 1: We can just have like a core and then we can have the model build out UIs and apps and services using these building blocks. So there's this like meta framework for building into the future that is not the same as automating your work in that you're not trying to get a model to manage all of your work

Speaker 2: and to spawn other models.

Speaker 1: Instead you are getting your model to use a specific set of tools to consistently build out the same types of applications. That way if you make changes in your core ui, all of the different apps that you've built with AI are going to reflect that change, which makes it very beautiful and much easier to scale out. LLMs, they like horizontal type work and the reason for that is they are great at doing surface level changes, but require a lot of validation.

Speaker 1: So if you take a model and you try to one shot a complex service with it, it's going to get lost like pretty easily. You're going to have a hard time keeping it on track. If you figure out how to automate it and keep it on track, it's probably just going to start doing unrelated stuff or reward hacking or giving up. So it doesn't make much sense right now, at least for a normal person to have like a 24 hour builder running.

Speaker 1: But it does make sense to have a core and then basically build out a bunch of Applications and then you are the validation step for all of that. Is that something you're going to want to do? Maybe not. I mean, it makes sense why you wouldn't want to do that, right? Like, reviews are the worst part of the job for a lot of developers. But if you're trying to build out a lot of stuff, that's kind of the only way you can make this work.

Speaker 1: In terms of the clis, I'm going

Speaker 2: to try to open a CLI here and let's open a new window and then bring this here and. No, we don't need that.

Speaker 1: So let's take a look at the app that I have here. So we have the Ghosty terminal. What I typically like to do is open like a few of these panes and I'll have these as long running processes. So if you go on my hugging face. Let's go over to hugging face and take a look at what I'm doing. You could see here that I have posted 72 models in the last six months.

Speaker 1: Most of this was enabled directly via using Ghosty. Having these panes open and then in each pane I would have like a process. One of them does observations. One of them runs jobs for like 12 to 24 hours. One of them actually ran for a week, if not longer. And we could see here that I can open the.

Speaker 2: Let's go back.

Speaker 1: Let's make. We're going to make GLM 5.1. I'm going to close these other ones out.

Speaker 2: So I'm going to clear this and I'm going to run a local model. So we're going to take a UI here. And where is the ui? Let's try to get it on the right side and close this.

Speaker 1: So I'm going to use a local model right now. So I just kind of want to demonstrate something. I want

you to build me

Speaker 2: a 3D application FPS game, Make it unique.

Speaker 1: I don't know, I'm just throwing random stuff out there. And please use the Agent browser CLI to validate that it works before you're done. So this is a fresh repo, there's nothing in it. And I will start this process and then what I'll do is like I'll have another one. And in this tab, let's say I'm doing something else completely. Like I'm maybe researching maybe.

Speaker 1: Yeah, like let's. Let's do the research thing so we

Speaker 2: can open a research tab. Let's see if this is going to

Speaker 1: work because I've been messing around with some stuff.

Speaker 2: Seems like we have a new version of Droid.

Speaker 1: But let's take a model that is

Speaker 2: pretty good at research. Let's take GPT 5.4.

Speaker 1: I want you to deep research how

Speaker 2: to build a 3D FPS game in the browser and create a scope of work.

Speaker 1: One thing I'm trying to show here is that so this is my home lab. So this is the model that I'm running locally. I'm running a pruned and quantized version of GLM 5.1. Then in the same harness, which is Droid, I'm also running like GPT 5.4. I can launch another one and we

Speaker 2: can take a look at this as well.

Speaker 1: Go here and I could just keep doing this with different models. I can have one review the other. I can have one be a sub agent, one like. So this is the GLM 5.1 directly from the ZAI thing. So make a subdirectory called don't touch

Speaker 2: and in it make a 3D FPS game in JS and HTML. Then load it in the browser for me and let's go check on the first one.

Speaker 1: So this one is still doing its thing, this one is researching and this one is just getting started. So what

you can see is that I have like I could just spawn all of these models and each one is good at something specific.

Speaker 2: Let's go back here and let's make another directory.

Speaker 1: What model would we want to try? We want to try Kimik 2.66. We're gonna did this. Okay.

Speaker 2: CD community. That's strange.

Speaker 1: Ah, so, okay, so we're gonna launch this and we're going to set up

Speaker 2: the QEMI model as well.

Speaker 1: And each one of these models is good at something different. I think GLM is really good at DevOps. I think Kimi is great at like 3D. It apparently has a very low hallucination rate, which is great if you're working in science. It's good at research, it's good at using sub agents.

Speaker 2: So let's go over here and then choose Kimi K 2.6.

Speaker 1: So the benefit is that you have all of these different models that you can lean on and each one of them has like its own skill, its own thing that it does well.

Speaker 2: So here is GPT. It's done.

Speaker 1: It's good at scoping stuff, it's good at researching, it's good at validating work of other models. It's great at coding. It's not really easy to talk to as you can See here, like, it's either too verbose, too much lists, too much information, or it doesn't give you enough information. And all this stuff like you can work around it by having more than one model. Hopefully this is done sometime today. It's going to be pretty slow because it's running locally now.

Speaker 1: How would this apply to your work?

Speaker 2: Right.

Speaker 1: If you are using AI a lot, you need to start thinking about a few things. How much?

Speaker 2: So thinking tokens, 6x increase.

Speaker 1: I just want to show this piece

Speaker 2: of data because it seems to be

Speaker 1: pretty important and not many people are talking about it. The models nowadays, they are thinking. So they're producing about 6x as much content as they were maybe three years ago. And while this is great, it's awesome, right? Because you have smarter models. These models can do more tasks, they're more capable. The flaw is that you are spending six times more money. And if you're using a dumb model, you're probably going to waste a lot of your time.

Speaker 1: And if you're using a smart model, you're going to waste a lot of money. And so we have to start thinking about like the economics of how much money is it going to cost to run GLM versus Clock. What can I arbitrage? Because if the compute is free right now, is there a way for us to arbitrage that compute so that we can build out more stuff so that when it gets more expensive, we can afford it?

Speaker 1: I think we should start thinking like that. And I think tackling this problem has like a few directions. Is one, your stack has to allow for model agnosticism. We shouldn't have to have like a single stack that works for a single model. That's like, I get, I understand why some people like it. It might be easier, but then you're just limited by the whims of the company that makes these models. And I don't feel comfortable with that.

Speaker 1: Personally, I would rather have like 10 different options and be very comfortable switching between them and compensating for their flaws than being stuck with Claude and then having Anthropic ban my account for some random reason.

Speaker 2: So don't touch. Let me see. This probably is the app here. Okay, so it did open the app,

Speaker 1: move click anywhere to start. So it says click anywhere to start, but clicking doesn't actually work.

Speaker 2: So I'm just going to send it

Speaker 1: that and see what it does. This is going to take a while again because it just takes a long

Speaker 2: time for local models to finish.

Speaker 1: And I think I don't have this as optimized as it should be. But it is going to finish in a minute. We should get used to working with stupid models. We should get used to tracking our budgets. We should favor applications and tools that allow us to bring our own subscriptions, our own tools into them.

Speaker 2: Google Chrome.

Speaker 1: I don't know why it open to.

Speaker 2: Okay, this is one. Let's see.

Speaker 1: Yeah, that's. That's very interesting.

Speaker 2: Okay, Quick Codex.

Speaker 1: And you can see they're all like doing stuff on my computer at the same time. It's pretty interesting. I think that these models have gotten to a place where they're really excellent. Like, any one of them could be. Any one of them could be Daily driver. Even something as small as 27 billion parameters is capable of daily driving and doing real work and is competitive with

Speaker 2: models that are ten times its size.

Speaker 1: That doesn't mean that, like, obviously it's always going to be that, but generally speaking.

Speaker 2: Okay, there you go.

Speaker 1: First person shooter. And you can see here the same exact issue that was with the original GLM is with this pruned and quantized version. They kind of.

Speaker 2: Okay, three, this is the eraser. Okay, this is not working.

Speaker 1: It's not working. It's not working. So. And we can tell, like. Yeah, I just wanted to demo this just to make it clear, like, you know, there's been a few things that I'm showing you. One is like how the style of the work that I'm doing and what I think we should do collectively, which is resist lock in.

Speaker 1: Two is kind of demonstrating where open weight models have gone over the last few years. And then three, showing you the difference between an open model and a closed model. So an open model that is self hosted, compressed, quantized, all that, and an open model that is hosted by the original maker of the model. And you can see that the difference is really not that much. Quantization doesn't not really hurt these AIs as much as people think.

Speaker 1: Yeah, this one is still not working. So hopefully this was helpful. Thank you for listening to my rambling and I will see you again in the next.