The Cloudcast (00:02.99)
Test test test. Three, two, one. Good morning, good evening wherever you are. Welcome back to the Cloudcast. We are coming to you live from the massive Cloudcast studios here in Raleigh, North Carolina. I hope everybody is doing well. I everybody's having a good weekend or beginning of your week whenever you're listening to this show. I am recording this outside the normal studio, so apologize if there's any sort of background echo. Not the normal room with all the...

things to reduce that. anyways, hope everybody's doing well. First off, I want to apologize for last week. We did not call out Mother's Day, which was by no means intended, just kind of slipped past us. So apologize to all the mothers out there. We are thankful for what you do, and we appreciate everything that you do. I want to dive into a topic that we've talked about quite a bit on the show. I feel like we've talked about quite a bit for the last year and a half or so.

you know, as AI was starting to become more mainstream, we were all getting used to what it meant, what the technology was doing and all those sort of things. And I think we were looking for, you know, what were some of the early, early use cases, early things that people were really grasping onto and that were starting to really start to take hold, especially within the enterprise in terms of use cases. And while we went through a number of them,

One of them that really kind of jumped out was sort of developer co-pilots or developer assistants, whatever you wanted to call them. You know, essentially, you know, the AI tools that were helping developers to write code, to better manage their code, to troubleshoot all sorts of things. And, you know, the initial feedback was, felt very good. You know, a lot of developers were excited about it. They were beginning to use it. I know, you know, every time we do a Cloud and AI News of the Month, Brandon Wichert talks about how he is a huge fan of it.

And we've talked to lots of people who are very bullish on it. And then as we started digging into it little bit more and more information was coming out, more people were beginning to use it, they were starting to put out some information because a lot of times, the question, at least initially, as many things do in our industry, it starts off as being very binary. OK, here's this new technology. Will it replace the previous thing? And in this case, it was, can these coding agents begin to replace?

The Cloudcast (02:21.294)
the thing that they are augmenting, which was developers. And that question, as with almost everything that we deal with in the enterprise, is sort of a silly starting point, but it's where we start. And then we started to see some numbers come out. And we were seeing numbers that were coming out from people doing survey work, people doing analysis of their early adoptions and so forth. And we saw a lot of 10 to 20 % productivity uplifts, 15 to 20%, maybe 25%.

Um, the numbers that weren't overwhelming, they were nice. Uh, you know, anytime you can get, uh, you know, a decent amount of, of productivity uplift, especially in a very short period of time. Um, it's interesting. Um, but you know, it also fell a little bit short of, you know, some of the proclamations from the technology providers of saying, you know, this is going to be 50 % uplift, 40, 50, 60 % uplift, uh, in terms of productivity. So, you know, we, we had, uh, we, had a initial moment of, yep.

This seems like something that's going to work. We've begun to see some numbers come out. The numbers weren't necessarily as high as people thought. But then I think we're still a year, year and a half, maybe two years into these sort of adoptions. And maybe if the uplift is 10 to 20 % on a yearly basis, then that becomes very, very interesting. We have begun to see some vendors, Microsoft in particular. Satya Dadela was doing an interview recently.

And he said, you know, somewhere between like 20 and 30 % of their code is now being written by AIs. And, you know, what he didn't really get into the details. People are curious. Does that mean like, you know, office 365 or, you know, power BI or something like that, or is that like, you know, internal stuff going on? Is that just CI CD pipelines and so forth? But at least it was him putting a, a number out there in terms of where they were and then starting to, kind of work off of that. So.

You know, again, we've seen some numbers out there. think we're also beginning, I think, to understand what does this mean? How should we even begin to measure this sort of space? Because development has always been a very difficult thing to measure. know developers have always been very particular about not necessarily wanting to use one specific measure to identify what they had done or

The Cloudcast (04:43.93)
put a value around what they did. In some cases, it used to be lines of code, and then it became other types of things. So this has always been an area that's been a little bit difficult to necessarily nail down exactly what, not what it's providing, but how do you best measure it, and how do you do it in a way in which you respect the fact that while writing code takes time and has become a very important thing, it's also something that

is a piece of the entire process of software. The other thing I think that we've sort of gone through, and this is where these numbers start to get interesting because we're already starting to see some companies not hire as many developers. In some cases, we've seen some layoffs, and it's hard to know whether that's seasonal or economic related, or if it's companies beginning to wonder, do I need as many developers?

You know, and it's a big change. feels like a big change from where we were, say, a decade ago or even five or six years ago when COVID hit. But, you know, going back to things like when Stephen O'Grady from Redmonk wrote, you know, Software Developers are the New Kingmakers, that sort of seminal book that was out there where, you know, I think we would go back a decade or so. were at the beginning of sort of this trend of digital transformation in which software was going to

eat the world, but it was going to define what your business experience was like, how your business managed data, how it managed collaboration internally, all sorts of things. And a lot of that's become normalized, right? It's become part of what every business has sort of gone through. Every company's had a decade or so to kind of do that. And I think to a certain extent, maybe what we're seeing is a little bit of also rationalization of how much do we have to own in-house versus can be picked up by a third party service, a SaaS service, if you will.

you how much should we be more efficient of the developers that we had in place? So I think all those dynamics are in play. You know, as we look at the technology itself, one of the things that we've often asked is, you know, how much should people be willing to pay for this? Right? You know, and this is, you know, kind of the classic economics question of, know, I'm going to invest in new technology. You know, what sort of benefit should I expect to get from it? You know, is 10 to 20 % uplift good? Is that

The Cloudcast (07:04.078)
10 to 20 % per year, is that 10 to 20 % over three to five years? And then you sort of manage the, you know, how much additional investment am I making for that 10 to 20 % or whatever the uplift is, right? And so we, you know, we sort of questioned as we often do during early cycles, you know, what, what should be people be paying for these types of technologies and then what should those companies be valued at? Right. And we've started to see companies like cursor, uh, who recently had a nine to $10 billion evaluation based on some funding rounds.

Recently saw windsurf was acquired by open AI for three billion dollars. So You know you begin to Sort of go. Okay, at least some numbers are out there. We're starting to see what people are paid for oftentimes the early Early adopters are going to pay a premium Above and beyond because they're trying to gain first mover status and so forth so, you know all those things are sort of going on and then we can overlay that with you know, the beginning of

seeing more and more adoption, more and more evolution of agents. So not just assisting developers, but beginning to see that next stage where you sort of ask what could be done autonomously? What could be done with fleets of sort of non-human AIs going off and doing a bunch of tasks? And we're seeing lots of companies launching agent frameworks. We're seeing lots of open source frameworks out there.

This is a space that is, it is rather interesting. There is, you know, economic value being identified. There is investment happening in this space. It's not exactly what people expected, but I think, you know, we are, we are going through a change of, you know, a previous sort of era in which we, placed a ton of emphasis on developers and, you know, again, those developers necessarily.

want to be measured in certain ways. They just wanted to be able to write code and do new things and be 10x developers. I think this is the normal rationalization that we're seeing. And we're also seeing the rationalization because we have a new generational technology that's bringing into question how fast should your overall software development happen? What should your software development, whether it's a factory or whatever it might be, how should that work?

The Cloudcast (09:23.366)
And we're beginning to have those conversations, think, taking developers out of the equation, not so much as like, should we replace them? But overall, what should that thing look like? What should the role of developers be? What should the role of AIs be? And I think that's the sort of next stage that we're beginning to get into. And I think that conversation is ultimately leading, you know, why we're seeing some of these high valuations, why we're seeing Google and Amazon and others.

you know, begin to put out rumors that they expect to be, you know, creating their own assistants here pretty soon and will be available in market. We'll probably see others as well. We'll see smaller startups, you know, start to be in this space. And so it really got me thinking, you know, where does this go next? Right. Is the next, is the next stage of this really just the playing out of, you know, widespread adoption of sort of enhanced developer productivity? Right. So do we just see this, this market sort of mature? Do we begin to see

you know, better understanding that, you know, these, these tools should be helping developers, productivity at say 20 to 25 % per year, right? Some, as, some sort of baseline, just sort of throwing a number out there. or do we begin to see this really take a different frame frame of, of, view, frame of vision, frame of opportunity, and again, begin to look at this as not just how many developers do you need or what kind of tools should you be using or what framework should you be using, but

You know, if software is still a differentiator for the business, but it's also something that is foundational to the business, you know, what should that look like? How can we be more efficient? How can we better, you know, leverage the ability to bring in innovation, you know, kind of some bigger picture thinking. And what really got me rethinking this space as opposed to just thinking like, you know, what will be cursor 2.0 or windsurf 2.0 or

Copilot 2.0, whatever that would be, I had a conversation with a couple of CEOs, CTOs recently, larger companies. And what they were laying out for me was their way of thinking about this. And what they were really thinking about this was they were saying, instead of looking at it from a increased productivity perspective, they were really asking themselves, if we started to take the limitations out,

The Cloudcast (11:48.844)
Right. If they said, how much software could we create? And what could that mean for our business? Whether it means new capabilities, new products, new ways to get into market. They were looking at it holistically. And they said, if we took some of the constraints out of it in terms of, forget about the number of developers we're dealing with. Forget about the cost of those developers. Forget about all of the interpersonal communications and the amount of overhead.

that implies when you've got multiple teams and multiple ways of doing integrations and all these sort of things, what could we start to think about? And they were really beginning to talk about this idea of instead of having multiple layers of traditional managers, team managers and project managers and directors or VPs overseeing a lot of different stuff and then all of the human interaction that goes along with that, as well as all of the sort human opinion that goes along with developers picking which language and tools and databases and whatever it might be. They started to say, hey, what if we started to think more like, you know, we have a set of people that are very smart, you whether these are project leads or senior developers or whatever they might be. You know, what if we began to give them resources that looked like fleets of AIs?

right, or augmented certain work that they did with fleets of AIs, right? And this is where we begin to see sort of that next sort of steps begin to evolve themselves, right? Because the first sort of baby step out of that thinking is, do you start to see specialized functions within developer workflows that are really targeted by a set of AI tools, right? Whether it's CI and CD pipelines, whether it's documentation, whether it's

security evaluations, bug fixes, long-term maintenance. Imagine, do you need to continue to keep very valuable people employed just to do long-term maintenance? Or could you begin to essentially offshoot that to a team of autonomous things just dealing with bug fixes and patches and regression testing and all that kind of stuff? So that was the first half step they talked about. The second step really became that thing where they said,

The Cloudcast (14:12.034)
number one, you know, we oftentimes have very, very smart people, who get frustrated. they are, they're very smart. They are senior technical people. They've done very good work over time. They've evolved to become again, again, senior dev, an architect, a team lead, whatever it might be. And they don't necessarily want to go into the management chain. They don't necessarily want to focus more on soft skills and their hard skills. And, know, they began to ask themselves, what if those people now had a new path and that path was.

you know, managing resources, managing allocation, managing outputs, but those outputs were more focused around agentic services, right? You know, sort of agents, if you will. And, you know, it began to sort of spark an interesting conversation of, you know, could that person manage 10 AIs? Could they manage 100 AIs? So that was sort of the first one because Nanyahu's starting to get into, you know, they can potentially do, you know,

Instead of doing eight hour days or 12 hour days, they're doing 24 hour days. Instead of having to worry about language barriers or worrying about, you know, personal opinions upon stuff or, know, kind of all the inner commercial inner communication and collaboration that goes on, you know, could that become just orchestrated work? Right. So that was, that was one thing they were starting to look at in terms of breaking down barriers or limitations that previously had been there.

right, that were human oriented, right? Whether they, it was where do we find people, how do they best collaborate, how are they measured, you know, how do we spend time, you know, kind of negotiating between their preferences of stuff, all those sort of things. And then the second thing became, you know, maybe it's not, like I said, it's not initially fleets, but then it became, well, you know, could they do more, right? Is this a good thing or a bad thing to start thinking this way? And,

The second piece that came along was, OK, so now we've started to think in terms of what's possible, having removed certain previous limitations that we had, which were things like, could we afford those people? Were those people in the right place? Could we find enough people that knew our existing code base versus this new code base that we had? So all those sort of things. And the next thing I started thinking about was, well, we

The Cloudcast (16:35.938)
Because there's so much innovation out there, there's so much technology out there, and oftentimes we can't take advantage of all of it because it's hard to find people to do it. It's hard to figure out how would we deal with this, what would our learning curve be, and if we have to kind of go through all those processes, are we going to miss the window to take advantage of this? And so they started to think about what would it look like to begin to do greater experimentation for things, right?

You know, oftentimes it's not hard to find somebody who went, a new thing came along. I spun something on my laptop. But you don't really get a chance to experiment with it because that learning curve plus all the possibilities just takes a good amount of time. so could, you know, what would happen if, you know, if you, if you think about project managers asking for things, you're building stories against, against whatever you're trying to build instead of getting, you know, one or two prototypes or one or two, you know, minimum viable types of things. What if you got.

10 or 15 or 20 or 100, sort of options at that, right? Those options often at times are very expensive. They take a bunch of time. They take a bunch of resources. People try and guess what's going to be best. But oftentimes, they guess what's going to be best based on the few options that are given to them. What happens if we begin to see greater experimentation? Because again, these autonomous resources can build options very quickly. As we've seen,

just using chat GBT tools, you can say, hey, write me something, but write it in a more humoristic way, in a more serious way, in a way in which it's going to be more attention grabbing. Well, you could imagine those same sort of things being, let's build this feature. Let's build this user experience. Let's build this one. So it's more geared towards a Gen X crowd, for us as a millennial crowd, versus one in which the behavior is going to look something like this.

And those can often be very difficult if you are dealing with human restraints of time and capabilities and all those things. But could you begin to do more and more of those and potentially either get to a better solution faster through more experimentation or begin to realize that you could further segment what you build to be more specific to catered to certain audiences? And so that was an interesting viewpoint they were looking at is,

The Cloudcast (19:01.614)
This isn't really a, again, a replacement of human beings, but really an augmentation of the possibilities of what could be for your business. So that was sort of the second area that they were really kind of digging into. The third was, do we start to see teams, companies, organizations begin to build out these skunkworks in the same way, centers of excellence, the same way we saw...

when people were doing this with building cloud native tools or DevOps or other things a decade ago, where it was like, this stuff is going to be very different from what we're used to, kind of in the Gartner bifurcated mode that they talked about, mode one and mode two, or bimodal. Do we begin to see companies building or creating or incentivizing somebody to go see what's possible with this? And I could very much see this happening.

in which somebody goes to one of the senior engineers who maybe is frustrated, doesn't necessarily want to be put down the management track and says, hey, I want you to take six months or nine months or whatever it might be. And I want you to go experiment in this space and see what you could create, see what's possible, see what systems would need to be put in place to make this real. So I'd be very curious to start to see if we get to see if we start to see those types of things starting to emerge. And the last thing that we really kind of discussed

Again, in the context of, you know, are things going to be, you know, what could be different if we started to remove some of those things? You know, oftentimes we'll talk to product managers and they will be, you know, they'll be frustrated because the number of things that they're trying to do isn't happening fast enough, or they don't have enough engineers to keep up with the number of requests that the customers are making of a feature or internal users, you know, want to make changes or, it be great if we could move into this new market with this new technology, but...

You know, we just don't have all the tools or anything, whatever is needed to get there. And it'll be very interesting to see if, you know, we start to get to this point where we can, we can build things faster. We can build more prototypes of things better or faster. can build more experimentations faster. What happens when that backlog goes to zero for product managers, right? Are they prepared to start to think faster about ideas? Are they prepared to take feedback from the marketplace faster? Are they prepared to.

The Cloudcast (21:23.054)

come up with directional variations that could happen in that space. So I think what was really interesting to me is the way that this next generation thinking wasn't looking at it purely from a replace humans or augment humans. It really was if we look systematically at building software and that software being for expanding the business, reaching new markets.

giving a more catered experience to things, all that kind of stuff. What does that mean in terms of our thinking? What are the barriers today that are holding that back? What are the things that we could begin to do if those barriers weren't there? And I think that combination of new way of thinking about it, along with these new tools that are beginning to emerge and the standards are beginning to merge around them and all that sort stuff, really starts to paint an interesting picture of potentially what's next for this.

you know, age for developers, right? And it gets you away a little bit from the, you know, developers should be worried for their job kind of thing. you know, that's every person that's working with AI has that, you know, potential kind of shadow hanging over their head or cloud hanging over their head. But I think when you begin to think about it in terms of not just, you know, binary one for one, you know, replace or augment,

but you begin to think a little bit more systematically. It does get into some very interesting thought processes. And I think it begins to, in the same way that we saw multiple decades ago where people were rethinking how automobiles were built or how factories were organized around end results, but optimizing in every single layer. I think we're gonna start to see that in ways that we didn't necessarily see when we were first talking about kind of...

The factory automation in the same context as developers But I think now that we have the ability to have autonomous systems in places that we didn't necessarily have the ability to do that or systems that could You know produce much much more much faster whether those are experiments or real life with Certain constraints being taken out of them. I think it becomes very very interesting. So anyways, just something to think about would love to hear people's feedback on you know where they think the space is going beyond just

The Cloudcast (23:51.896)
like I said, cursor 2.0 or co-pilot 2.0. We'll be very interested to hear, you know, kind of where your thought processes are, where your companies are with this stuff and definitely an area that we'll be continuing to cover on the show going forward. So with that, thank you all for listening. Thank you for telling a friend. Thank you for helping us spread the word about the show and get it out to new communities. And with that, I will wrap it up and we'll talk to you next week.