

Talks at GS

The Future of Software Engineering: Cognition's

Russell Kaplan

Date of recording: July 18, 2025.

Russell Kaplan: In the coding sphere, the first popular assistants were really the co-pilots. So while you were writing code, getting suggestions in real time for how that code should look, how it should read, how you should finish it. Now we're at the precipice of the AI agent era where these are true autonomous decision makers, where you're delegating work, and then the agent is going off and doing it. What I think this means is that coding, math capabilities, domains with formal verifiability are going to continue to explode in capabilities.

Marco Argenti: Welcome to another session of Talks at GS. I'm Marco Argenti. I'm the chief information officer at Goldman Sachs, and today I have the honor of being joined by my friend Russell Kaplan, who's the president of Cognition, an artificial intelligence startup best known as the makers of Devin, which is, as we like to say, the first AI software engineer. So welcome. It's great to have you here.

Russell Kaplan: Thanks so much for having me.

Marco Argenti: So a little bit about your history, your personal history. So before joining Cognition, you were at other AI companies. You were at Scale AI. I think they bought your computer vision company.

Russell Kaplan: Helia.

Marco Argenti: Helia, great. And you worked at Tesla as well, right?

Russell Kaplan: I did, yes, building the auto-pilot stack on the sort of computer vision side.

Marco Argenti: Great. And then these days, this probably should not surprise anyone but Cognition reached a \$4 billion valuation in March, up from 2 billion the year prior and I guess up from zero the year before, right?

Russell Kaplan: Yeah. We were founded in January of 2024, so it's been a fast-growing journey for us.

Marco Argenti: Wow. All right. So it's great to have you here. Tell me a little bit about what Cognition is focused on but also what is the problem that you guys are really trying to solve for the industry and also for your clients like GS?

Russell Kaplan: So when we started Cognition, it was originally started as a pure research lab. We were focused on a single question, which was: What does the future of software engineering look like? And many folks on the founding team knew each other from the competitive programming community. There is kind of an Olympics for coding. Every country sends four people a year, and you compete in these Olympiads. And many folks on our founding team were the sort of gold medalists from the United States, all coming together to figure out, okay, how do we actually solve software engineering?

And as we did more research, we realized that it was becoming possible to build autonomous coding agents. So not sort of a chatbot system but a true end-to-end agent that you can delegate complete units of work and get back a full result. Not just the code but code that's been tested, that's been iterated on, and that's now ready to be merged

into production systems. And that's what we released in March of last year.

Marco Argenti: So I think what I've seen in the last maybe six months has probably moved my expectations of what AI can do more than what I've seen in the last, like, two years because, at least from a getting closer to kind of action being really useful and impactful, I think there was a point in time where switching from Q&A, switching from, you know, "I cannot give you more information," to, "I'm actually doing things for you," has been a game changer. And that's also what triggered our conversations and eventually our choice to kind of work with each other.

In your words, how would you explain to a potential customer or someone that is looking at us today, the difference between an assistant and an agent?

Russell Kaplan: So assistants were sort of the AI tools that are now very popular that we all know and love. For example, ChatGPT. You have a chatbot you're interacting with and you get a real-time response. And it's incredibly helpful. And it's been amazing to see how far assistants have come.

In the coding sphere, the first popular assistants were really the co-pilots. So while you were writing code, getting suggestions in real time for how that code should look, how it should read, how you should finish it. Now, we're at the precipice of the AI agent era, where these are true autonomous decision makers, where you're delegating work and then the agent is going off and doing it. And it might take tens of minutes or hours. It's a different paradigm that's asynchronous instead of synchronous, but the results you get back are orders of magnitude more complex, more rich, potentially more useful. Which, I mean, you have a lot of experience with using these types of tools yourself.

Marco Argenti: Well, I have to say that I wrote more code in the last probably three months than in the last 15, and most of it I haven't actually written it. For example, I have this recording studio, and sometimes engineers come in and they want to use stuff. And so what I've done in a weekend, literally, I asked Devin to build a studio AI. And I decided to do it the right way. This is something that it needs to live in its own website, its own URL, it has to have its own knowledge base. I want to use CI/CD. I want to

deploy through Vercel so it's automatically triggered when I commit something, and all the good stuff that it would have taken me, I don't know, maybe a month to build. And literally it was, like, I counted my own hours, it was maybe, I don't know, ten hours on a weekend. And it was just incredible.

I asked at one point to put a model selector, and for some reason Devin decided at the end to put a little bit of a vanity line in the site. So when he added the model selector, when you select the model, I noticed that he automatically changes the string there to powered by the model that you select there, which is a little nuanced but I could have forgotten that. A developer could have easily forgotten that because one was a static string, like almost like a copyright. And those little nuances are kind of what makes me think, wow, this is not just a set of rules in the back and, like, an expert system old way. This is really something that has some thinking capabilities. And then sometimes it kind of does things that do not even get you in the fastest way possible to the objective, but it kind of does things that might be superfluous but really, really good in terms of the overall user experience. So that's my experience there.

Russell Kaplan: I think it's super interesting to kind of hear you also talk about your high-level picture for the tech stack for that project because, in many ways, this is what we see software engineering moving towards, which is every individual engineer becoming sort of a tech lead manager of a fleet of AI agents. And so you might have some high-level vision of, "Here's how I want this project to be architected," but then a lot of the execution details can be delegated to AI. And that's actually one of the biggest sort of skills we're seeing correlate with really strong success in this AI agent era is the ability to break down complex projects and delegate them into units of work for AI agents.

Marco Argenti: How do you kind of guide Devin to essentially "do not regress" and to kind of keep on rails in a way? What is the thinking in terms of how do you contain this power and this intelligence?

Russell Kaplan: Yeah, that's kind of the central question for the folks like us who are building these autonomous agents, which is: How do you keep them on the rails for longer and longer periods of time, more and more complex units of work? And sometimes if the task is so simple,

maybe the asynchronous paradigm of launching an agent and going off and doing something else, maybe that's not the right one.

And we can talk about this later, but we had a big week as a company this past week. We actually acquired another company called Windsurf, which builds the sort of first agentic IDE, and that experience is much more human in the loop throughout the creative process. And it's something that we saw as maybe the right experience for tasks like that.

Marco Argenti: So does it mean that you guys are kind of losing a little bit of your differentiation? How do you stay true also to your own original mission?

Russell Kaplan: The way we think about it is that we want to build the future of software engineering, and one of the most important parts of software engineering is using the right tool for the job. And right now, AI has progressed to the point where we can have these autonomous agents, we can have Devin for certain classes of work, right?

The way I think you've described it and I've heard others

describe it is think of Devin like an army of junior engineers. They still need some steering. They still need some guidance. They still need some scope. And so there's a class of technical problems that can be broken up into that sort of army of AI junior engineers. But there's also a class of problems where we're not there from a capabilities standpoint. And as machine learning continues to progress, I think we'll get there.

But in the interim, we want to have the most complete solution in the market. And I think for engineers who are building systems, sometimes you have to have high leverage and sometimes you to have be really in the detail.

Marco Argenti: Are you guys focusing primarily on sort of the vibe coders of the world? Or are you maturing being more of an enterprise? Tell me a little bit about how you balance this a directive of growth to come.

Russell Kaplan: Yeah, when Cognition got started, we made the unusual choice for a startup to work with big companies first because we found that the engineering problem of writing code in an existing code base is a lot harder and a lot more complicated than building things

from scratch. And so an organization like Goldman Sachs, there are tons of code, proprietary frameworks, proprietary programming languages, so that was a research problem that we really sunk our teeth into. And as a company, we've been primarily focused on the enterprise.

At the same time, the future of software engineering might involve way more people creating software before than people have in the present. It used to be that you have to sort of learn this kind of arcane skill of how to write code with very specific syntax, very specific technologies. And we see a future where there's way more software engineers than there are today, but the nature of the job has changed.

Marco Argenti: And how does it change?

Russell Kaplan: I think it looks something like -- well, actually, maybe just briefly on the sort of history of software engineering. You know, we've been trying to interact with computers and tell them what to do for a long time. I think the 1890 census was the first one that used punch cards. It was on time and under budget because of punch cards as sort of the first data collection at scale

mechanism. And then we had Assembly in 1947 and Cobol later. And the whole history of software engineering is greater and greater abstractions but really with the same goal of just telling your computer what you want it to do.

And we think the future of software engineering is that everyone can tell their computer what they want it to do. And so with Devin, we've still been focused on working with software engineers and tech-adjacent functions because, as you know, these AI systems aren't perfect today. And so if they're 80% or 90% of the way to a solution, you want to be able to step in and give that feedback for the last 10%. But as they get better, more and more folks are getting empowered.

And I'll give you one funny anecdote. We built Devin as a software engineering, but our operations lead realized that she could run our lunch catering through DoorDash with Devin because Devin can make long-term plans, it has a browser, it has a secrets management system to handle credit cards, and so we started getting all these Devin DoorDash orders. And we looked around like, "Who's doing this?" And our operations lead, who doesn't know how to code, said, "Oh, yeah, that's me. That's Devin."

Marco Argenti: Does she still have a job?

Russell Kaplan: She does! In fact, she's, like, maybe three times as productive as I would have expected an operations lead to be, maybe in part because of these tools.

Marco Argenti: So are you seeing Devin involved beyond software engineering?

Russell Kaplan: I think if you can solve software engineering, you can do a lot more, too. And one way to think about this is what does it take to really solve software engineering? It's not just writing code, right? I mean, you've said this. What percent of the job of a software engineer is writing code? It's less than half probably. There's what you plan, there's how you interact with your colleagues, there's how you decide what to build. And then when you write the code, you have to sort of test the code and run the code. And I think there's a world where those agentic baseline capabilities get really useful in a lot of other domains.

Marco Argenti: But through code? Or also you think

it's just, like, at the end of the day, it's just logic that then takes a different form or process? For example, it could be a policy. It could be something that essentially drives some action and is executed somewhere? Or is the artifact, in your opinion, primarily code?

Russell Kaplan: Right now for us the artifact is primarily code, but the underlying system capabilities that are needed to do that well are reasoning, not code per se. And I think as we build more and more sophisticated systems and we get better and better at reasoning, you can imagine code plus using your computer, code plus reasoning about the documents and the documentation of your organization becoming really relevant.

Marco Argenti: I've noticed that the capabilities of Devin increase kind of are in lockstep with the release of new models, right? And very quickly. So when Cloud 4.0 was released, you know, like, almost like the day after I felt that Devin was getting much smarter. So do you want to tell us a little bit about your relationship with some of the model providers and how you keep in sync with sort of the engine that is powering your car?

Russell Kaplan: Yeah, we're in one of the sort of greatest technology acceleration waves ever maybe with these models getting better very significantly every few months, every year. There was recent research from an independent lab that found that, on average, these AI agent capabilities are doubling every seven months. Meaning that, if you look at the complexity of a task that can be handled by an AI agent, the time it would take a person to do that task that the agent can now do, it doubles every seven months. And so if you extrapolate that trend, that's significantly faster than Moore's Law. And so definitely an input to Devin and underlying model capabilities improvement.

At the same time, we found that these models were sort of built originally as these text completion engines. Trained on the whole Internet. Figure out the next word. That's the root of how these models are trained. Versus now, the tasks we're trying to do, it's very different. It's to complete real-world projects. To complete units of work. And so we do a lot of our own research internally as well on how to build the best agent where some of these models are useful inputs.

Marco Argenti: Amazing. I think maybe I'll ask you one last question before closing, which is asking you to literally predict the future a little bit, which obviously in a time like this is probably the most difficult thing. So what do you see coming? If you had to picture, you know, like, we have the next Talks at GS maybe in six months or in a year, what are some of the things that you think are on the horizon at this point?

Russell Kaplan: Yeah. You know, one of the reasons I started working on Cognition was, when I was at Scale, we oversaw the data pipelines for all the foundation model labs. And so every new model release is a few months downstream of the data that's going into those models. And what became clear is that these models are getting better at everything, but in particular they're getting better at domains where you can automatically verify the quality of your work. And that's one of the reasons that coding has progressed so fast is that you can write code, but then you can have your computer run the code and you can see does it compile? Does it pass the test? Does it not?

So I think to predict the sort of next 12 months, it's really helpful to think about what are the domains and what are

the infrastructures that we've already set up where you can have that automatic verification loop? You know, for the machine learning researchers in the room, that can then be used for reinforcement learning to improve even faster. And so what I think this means is that coding, math capabilities, domains with formal verifiability are going to continue to explode in capabilities. And very pragmatically, I mean, our research goal is to get Devin promoted to senior engineer.

Marco Argenti: In the next, say, 12 months, what do you think is going to be the biggest constraining factor? It used to be data, but now everybody's getting so good at synthetic data. What do you think is the bottleneck?

Russell Kaplan: I think the real bottleneck is actually going to be power. Energy production in the United States is much slower than it needs to be. I think we probably hit very serious bottlenecks. Maybe not in the next 12 months, although it's already starting to happen. Maybe in the next two years because these systems are enormously compute hungry and they have the property that, the more compute you add to them, the more capable they become. And so I think if you sort of look at the trend lines of how

much more compute we want to use as a society and how much more power is available to power that compute, we're about to hit, you know, maybe not in the next 12 months but maybe not long after, we're going to hit a serious wall.

Marco Argenti: Interesting. Well, Russell, thank you for being with us and congratulations for your brilliant career so far.

Russell Kaplan: Thank you.

The opinions and views expressed herein are as of the date of publication, subject to change without notice, and may not necessarily reflect the institutional views of Goldman Sachs or its affiliates. The material provided is intended for informational purposes only, and does not constitute investment advice, a recommendation from any Goldman Sachs entity to take any particular action, or an offer or solicitation to purchase or sell any securities or financial products. This material may contain forward-looking statements. Past performance is not indicative of future results. Neither Goldman Sachs nor any of its affiliates make any representations or warranties, express or implied, as to the accuracy or completeness of the

statements or information contained herein and disclaim any liability whatsoever for reliance on such information for any purpose. Each name of a third-party organization mentioned is the property of the company to which it relates, is used here strictly for informational and identification purposes only and is not used to imply any ownership or license rights between any such company and Goldman Sachs.

A transcript is provided for convenience and may differ from the original video or audio content. Goldman Sachs is not responsible for any errors in the transcript. This material should not be copied, distributed, published, or reproduced in whole or in part or disclosed by any recipient to any other person without the express written consent of Goldman Sachs.