

05202025 Microsoft Build Keynote Charles Lamanna

## **Microsoft Build 2025**

**Charles Lamanna**

**Ryan Cunningham, Farah Shariff, Shilpa Ranganathan**

**Scott Guthrie**

**Tuesday, May 20, 2025**

**CHARLES LAMANNA:** Good morning, everybody. It's great to be back at Build. It's always exciting to get time together with all of you. As you saw from Jay's session, there are going to be a lot of agents created over the next couple of years. Developers are going to be building agents. Companies are going to be deploying and adopting agents, and they're going to be the new building block of IT and apps and services.

And the scale is staggering. IDC estimates 1.3 billion agents are going to be built by 2028. That's just a couple years away. And the reality is all these agents are going to have lots of expertise, lots of skills and lots of capabilities. The big question is going to be how we navigate all these different agents.

That's where Copilot comes in. It's one place to find and use all of your agents. It extends across all of your work. It follows you from app to app. It makes you more efficient and puts all these agents right at your fingertips.

It's not just going to be Copilot. We also are going to have to collaborate and communicate with these agents, which we'll do right inside of Microsoft Teams, the same way that we communicate with other people on our team. And we're also going to have to reimagine applications in a way where people are using the GUI, but also agents are in the background and using that same user interface.

As we build all of this, we're going to have to secure it. And the good news is your existing security capabilities will keep working: Entra for identity, Defender for protecting your back end and all your systems, and Purview for data labeling and data protection. And to see how all this fits together, we have a whole bunch of demos to show you.

And I'm going to start by showing Copilot in action and really highlight some of the things that you heard from yesterday, from Satya, this idea of a 5 in 1 experience in Copilot. So, if you look on this left-hand side here, you have search, you have chat, you have agents like researcher, analyst and more, and even things like notebooks and pages.

So that way you can go to one place and always get access to all the AI capabilities you care about whether the agents are built by Microsoft or built on the back end by developers or other companies. And one of the things that I always use Copilot for is research.

I got a message from Jay. Let's see. AI never sleeps. It's a new hour; it's a new paper. So, Jay's asking me to go check out one of these papers about some new RAG approaches for agents and

back ends, and compare it against what our team is doing. So normally this would be a time when I feverishly start reading through the paper, but I can use Copilot and the researcher agent to get caught up and not miss any of the important details really quickly. So, I'll come in here and just paste my prompt and I'll reference the document that I care about.

I zoom in here, you can see something really powerful about Researcher. It knows my documents, my meetings, my files, the specs, the arc diagrams. It even connects back to GitHub or Azure DevOps to compare this paper against our internal generative strategies. And this is what's great about researcher: it knows how we work. It's not just going out to the web when it generates the response.

And Researcher, like any good Researcher, doesn't run off and start doing its tasks. It has some follow-up questions. How long do you want the paper to be? Anything to pay attention to in particular? And the big thing I care about as a developer is how it diverges from the current approaches that we're using.

I'll send this message and what's happening here is this going out and doing that comparison for me and figuring out what we're doing today. That's not easy. I didn't write it in the prompt. It has to go to all those data sources I talked about earlier. What the Researcher agent does is execute a series of steps.

This can take ten minutes, 15 minutes or even longer to come back with a response. I won't make us wait for that to come back, so I'll switch over to one already completed earlier. And what's great is you can see it started to do this breakdown for me, comparing multi-agent RAG to our own internal generative answers approach.

The bit that I really like about this response is it's pretty thorough, very verbose, but there's a hero table that I really care about. And what that does is it clearly separates what the paper does against what we're doing today. So, I can look at query handling, retrieval sources or doc selection.

With so much research happening, so much change happening, I can use this to figure out the right algorithm and the right approach easily. And this isn't going to replace what you do inside of GitHub Copilot or VS code, but this is more about education and research and preparation. If I come down here, you can see all the amazing sources that were pulled together.

To make my life easy, I'm going to create that page which pulls together that researcher's response, and I'm going to take this and just share it right back with Jay. And this makes it so I don't have to type up anything myself either. And this is going to be that new way of working. Agents make us more efficient and more productive. And if I want to, I can go deeper right into that experience.

Now, if I come back over here to Copilot, it's not just about Researcher. We also have a huge collection of agents available inside the agents store. We have hundreds of different agents who can pick whether they're developer or sales or finance or more. And the agents you build can

also show up here. And that's something that we're seeing change pretty dramatically with agents is this open ecosystem.

We're all going to have our agents that we like and that we use based on our developer environment. And we're not going to change that support here at Microsoft. And a key part of that is MCP or the Model Context Protocol. As Kevin said yesterday, that's just like HTTP. This is the new way these agents are going to communicate back-end systems and work with all the different things that we care about.

And it's not just about MCP clients, us building things like Copilot or GitHub Copilot, but also MCP servers. And we're excited that we published Dynamics-365 MCP servers available to anybody to use. And we also announced support for agent to agent or a two-way protocol from Google. And we're going to support all open protocols that matter to our customers and to our developers, because this is going to be an open Agentic web as you move forward.

This ecosystem is already flourishing. There are hundreds of partners already adopting and integrating through graph connectors, agents and more. Companies like UiPath and Gong are showing up with first class experiences inside Copilot and agents. So, no matter what IT environment somebody has, these agents will work great.

And three of the most powerful examples that we have are around Adobe, SAP and ServiceNow. These are some of the most important applications in any enterprise environment. And they're going to be available right inside Copilot with AI rich experiences. Let's hear it from them in a short video.

**VIDEO SEGMENT:** AI agents are really going to change the way we work.

Take a look at the intersection of Microsoft productivity users and SAP's business application users. It makes a lot of sense to give them the ability to get more done.

Adobe Marketing Agent and Microsoft 365 Copilot together unlocks so much. Marketers don't have to switch applications now to make informed decisions.

ServiceNow AI Agent and Microsoft 365 Copilot unified in one experience means your AI agents can pull context from Teams chat, Outlook calendar and files.

Marketers instantly access audience analysis right in the applications they use. This is the next level of data driven decision making.

Choosing Microsoft Copilot to augment the capabilities of SAP tool was natural for us. It's the first step in bidirectional agent to agent collaboration.

We're shifting AI from a tool you use to a system that you collaborate with.

This is a perfect example of where the future work is going.

**CHARLES LAMANNA:** So, lots of agents. That's kind of the ongoing theme. And one of the challenges is you don't just want only Copilot to be the place you think about bringing order, we're going to have to create some structure around these agents too. And to do that, we go and draw from past experiences.

How do we organize teams of people working together? You have groups, you have permissions, you have working spaces. We're going to do the same type of thing with agents appearing right alongside all of our organizations. This combo of agents and people working together, this is going to be what the modern team looks like. And what that means is we're going to have to work through a new way of getting our jobs done.

Agents are emerging at massive scale. The applications that we use every day, the GUIs, the terminals, everything is going to have to change. And the way we collaborate is going to evolve. All these things require new tools, new thinking, new ways of getting our job done.

The first big component of that is going to be Copilot Studio, an incredibly easy way to create agents or PowerApps, a way to rapidly build out these new types of intelligent application experiences in Teams that is growing and evolving to support these teams of people and agents working together.

Wherever people are working, you will have access to all this great agent capability that we build and that you build. And if I look at Copilot Studio, we really view it as the easiest way to go create those agents. It's used by over 230,000 organizations and it wraps up the models, the connectors, the orchestrators, the observability, diagnostics and more.

So, you can have a one-stop shop and spend more time building, as opposed to managing the agents that you create. This is best seen in a demo. So, I am super excited to welcome Ryan to the stage to show it all in action.

**RYAN CUNNINGHAM:** Thank you, Charles. CSX is one of the largest transportation providers in the country. Their train division alone operates about 20,000 miles of track. And what you're seeing behind me on the square screens is a dramatization of the type of monitoring a team like that needs to make sure such a complex system stays running smooth like butter, because things happen, you know, and you want to be able to monitor just in case any incident might occur.

There's one now, in fact, about 24 miles outside of Bardstown, Kentucky. A freight train has had to come to a stop because of signal failure. Now everybody's okay. But this thing is carrying class 3 hazardous materials. It's blocking a main line. And by the way, it's 3 a.m. in February and it's snowing.

We got work to do, we've got trains to reroute, crews to dispatch, regulators to notify. That is a lot on the shoulders of the ops team working the night shift at CSX. And it's a perfect opportunity for agents to jump in and help out. And we're going to build them in Copilot Studio. Let's take a look.

Now, for those of you who have not come face to face with an agent in Copilot Studio before, this is what it looks like. It starts with plain-spoken, natural language instructions about what we need this agent to do and how to react, and a grounding in the knowledge from our organization.

Here are our policies. Here's what happens when an incident occurs. But folks, these agents in Copilot Studio now are not just waiting for somebody to ask them a question and answer it on a document. This isn't just a chatbot.

Agents also have tools and skills to do work inside of the organization. And because Copilot Studio is part of Power Platform, we have more than 1,500 connectors out of the box, everything from Acumatica to Zendesk. If it's got an API, you can work with it with an agent.

As Charles mentioned, we're also now supporting new open standards like Model Context Protocol. To be able to go hook up to MCP servers. And here's one that I've created inside of CSX to work with some of our compliance documents. I can authenticate it and pull through all of those skills and capabilities, keep them automatically up to date.

I also want to give this agent skills and tools of its own. And one of those important ways to do it is with a prompt. I can create very custom prompts. I'm going to paste in a pretty specific set of instructions for how I want this agent to react when a new incident occurs and I'm going to select now from any model at my disposal.

In fact, I've got a version of the DeepSeek reasoning model that I've fine-tuned and post trained in Azure AI Foundry. And now, like any AI model in Foundry, I can bring it here into Copilot Studio and see how it's going to perform in my agent. And that means I can have my most advanced AI developers and data scientists do that foundational work and then consume it up here in Copilot Studio for hundreds or thousands of agents.

And so, you see, DeepSeek took a minute to reason over that and brought back a pretty verbose response and did exactly what I asked it to do in the prompt. And I can use this really high productivity test to harness, to iterate and get it to do exactly what I need.

Now, the thing is, not everything in our modern organizations yet has an MCP server or even a Rest API. And that's where another really important tool for agents is the ability to just use computers directly. And we're really excited to have computer use now be part of Copilot Studio.

I've set up a computer use capability here. I've authenticated it already to a virtual machine that's running some software inside of CSX. And I've provided some very plain-spoken instructions about how to use that computer. Go to this website, open a certain screen and do some work.

What's cool is that I can just watch right inside of this test, my agent fires up and gets those things ready for me. It's going to connect with the virtual machine. It's going to initialize a session. And then pretty soon it's going to start using software. I mean, this is night and day

different from things like RPA of the past where I would have had to script and granularly record every single click and hope that nothing changed.

Now the agent can use a computer just like a person can. But look, those are all very cool tools for agents, but do you want to know what the most powerful tool for an agent is? It's another agent. And now Copilot Studio supports multi-agent orchestration.

And so I can come in here and add agents -- not even just other Copilot Studio agents -- but agents from Fabric, agents from Foundry, agents using the Microsoft 365 SDK. And I can build a team of agents to work together.

What do you think? Should we send this team to work? Should we see how they react to that incident we just received? I'm going to put in just a little bit of information here about the incident and we're going to see these agents get to work in real time. Let's see how they do.

Sure enough, we're already filing a report. And then the inspection agent is going to pick it up and go look for information about that locomotive. The crew management agent is going to wake somebody up in the middle of the night and get a crew dispatched. And the route availability agent is already trying to figure out how to reroute trains.

Now that is a pretty fast response. What do you think, Charles? I think we're going to get this thing back on track in no time.

**CHARLES LAMANNA:** The first demo ever that involved a train crossing. So, thank you Ryan. You think about these agents, they're working in the background. They're helping with the scheduling, the dispatching, the workers out to look at the rail cars. But you also are going to want apps in the foreground that people and agents can use together to get the job done, which means you're going to need a new type of application experience.

So PowerApps makes it easy to do this. It already provides that full stack experience: UI development, middleware, connectors and a data platform all in one integrated SaaS offering. But now you can layer on agents to that same secure platform and these agents can work with the apps and in the background. So, I hand it back to Ryan to show us in action.

**RYAN CUNNINGHAM:** All right. So, agents are off doing work, but what does that mean for the experience of the humans? To take a look at that, I'm going to put on my operator hat. We're going to go into the control room at CSX and look at a whole new type of application for working with agents.

Now, this is not your uncle's power app. It is not just about manual data entry. In fact, a lot of the agents are doing that work for us. And now I'm looking at a feed of activity of what those agents are doing.

And I'm playing the role, in fact, of a regional agent manager at CSX. I have a number of agents that are under my supervision and I can zoom in and see all the work they're doing

autonomously on my behalf and I can spot check it and make sure that they're on the straight and narrow.

Look, agents will not do 100% of the things 100% of the time. They will also need my help. And here's where I can zoom in to where the agents need assistance. And in fact, I can see here exactly the chain of thought and process that an agent took and where it got stuck creating an incident. I can drill right into that particular record and get this agent unblocked quickly. And that's why it's so important for this to be inside of an application.

Unblocking agents or correcting them is only one type of work. Agents can also notice trends and point out things that I might not even notice myself as a human and I can partner with them inside of this application to go investigate. Let's only look at the most severe incidents and let's show them in the southern states and show it to me by month.

I can be very plainspoken in my interactions with the agent and not only will it think about that and think about the best way to filter that table, but it will also dynamically generate user experiences for me to explore that question. That chart never existed before. No developer had to write the front-end code with D3. No data scientist had to build a cube. It's just automatically generated dynamically right at runtime.

Now, that begs an interesting question: what is the role for us as developers in this world where agents can generate user experience on the fly. And for that, I'm going to switch back to my developer hat and we're going to go use PowerApps. This is a very different type of PowerApps experience than what you may be used to in the past.

We have completely reimaged PowerApps from the ground up in a world of agents first. And so I'm going to start with a prompt here that really doesn't have anything to do with tech. It starts with our business problems and our challenges, just like I would send an email to a technical team.

In fact, that's exactly what we've created in this new capability for building a plan. You'll see we have a digital team of agents built right into PowerApps that function as an innovation team. The agent picked up that prompt and started thinking about who the users are, what jobs do they have and what are their user stories. And that's one of multiple agents that I'm live-coauthoring with right here in line.

An Access agent picked up and started mapping from those user stories how an actual process would flow, where would the AI agents work and where would the humans work. Now, of course, all of this is something I can edit. I can add new steps; I can drag and drop. I can partner with the agents at every step of the way, but it becomes an important foundation for building software. When I'm happy with the requirements and the processes, agents will generate a data model for me.

In fact, in seconds here, the data agent has picked this up and built a complex relational data model over Microsoft Dataverse, the same platform we run complex multi-billion-dollar CRM and ERP estates on top of. This is the kind of scale and sophistication that a vibe coder could

only dream of at my fingertips, working with agents in real time. Of course, when I'm happy with my data model and I'm happy with my stories, a solution architect agent will go recommend all of the individual pieces of technology for me to build.

Now, the cool part about all of this is it's not just an abstract plan. This is Power Platform, so I can take these assets that we're recommending and actually generate them in seconds. As a developer, let's build that app for safety incident tracking, fire it up over the data that the data agent generated and in a few seconds, see that app working in real time, pulling through all those great AI features and all of the context from the plan.

In fact, even agents are here with me as well, ready to go, helping me move this process forward. This is how we're going to do it, folks. 1.3 billion agents. I mean, we saw some amazing tools from Jay's team this morning about how we're going to lay the foundation for that and how we're going to do the hardest parts of the problem, but for that size and scale, for the developers who wear all kinds of other hats in the organization, this is where we need platforms. It's where Copilot Studio will help us build the agents, where PowerApps will help us build the experiences and together we're going to change the way the world works. Thank you all. Back to you, Charles.

**CHARLES LAMANNA:** Awesome. Thank you, Ryan. What we see is collaboration is going to happen in more places than just inside of applications. We're also going to have Teams being the place where we work together and it's the best collaboration space that we already know and love. Today, 320 million people are already using Teams every single day and there's going to be a new set of hundreds of millions of agents that will also be showing up inside of Teams.

It's the number one place for collaboration between people and agents as we move forward. Whether it's in a chat or in a channel or in a meeting or in a call, you will have this ability to very easily interact with these AI agents. As we make that easier for developers, Teams AI library is the missing piece. It makes it easy to bring your agents that you build and host wherever you want, with whatever model and whatever backend infrastructure that you want to have show up right inside of Microsoft Teams. That makes it easy, so all the AI agents have one place to go and collaborate. To show that in action, I'm excited to welcome Farah to the stage to demo it end to end. Take it away, Farah.

**FARAH SHARIFF:** Well, thank you, Charles. Well, I didn't bring any cool hats for my demo, but I'll show you some cool stuff. You just heard Charles describe Teams as the collaboration space for you to interact with both your coworkers and your agents together. Let's jump in and see it in action. You can see that the agent that Ryan built earlier has shared the new crew roster schedule in this Teams group chat, and in the chat, you can see that along with my coworkers. I also have the agent in it.

This allows us to access critical information right here in the flow of work, whether that information is coming from people or agents. Now, let me join my Teams meeting, my daily stand up, where I talk about the most pressing issues with my fellow engineers and see how agents can help. Hey team, thanks for joining me at Build. I'm clearly late to this meeting, but



you can see that the notes and actions from the discussion so far are already getting populated here, thanks to the Facilitator Agent, a pre-built Microsoft agent.

Again, along with my coworkers, you can see that I also have a Stand-up Agent here. I'll show you in a second how I built this agent, but this is essentially here to help bubble up topics for us to discuss, including topics sent over by other agents, such as the incident from earlier today, which was sent over by the Incident Response Manager Agent. It uses the A2A protocol to communicate with other agents, whether they're built in Copilot Studio, Azure Foundry, Teams AI library and more. You saw that I didn't even have to add the incident agent here, it was just able to talk to my Stand-up Agent.

Let's switch gears and let me show you how I built the Stand-up Agent using the Teams AI library. Just like you saw earlier with Copilot Studio, you can also build agents capable of using the A2A protocol and MCP using the newly updated Teams AI library. The AI library is an SDK that essentially takes care of a lot of the heavy lifting on things like authentication, conversation, complexities of bots, integration with AI models, etc. The result is not just faster development, but also more capable agents that are optimized for the human-to-agent collaboration inside Teams and works across your channels, chats and meetings.

Here's the Stand-up Agent that I was using in the meeting just now. As you can see, it took about 50 lines of code to get it all up and running and just one line of code to enable the A2A protocol. Now let's say that I want to use MCP to also enable my agent to access the train schedules. Well, again it is just one line of code to do that. That's because the Teams AI library handles the orchestration with the MCP server, regardless of your AI model. It is that easy. With the right SDK, you will always stay on track and never get derailed. Back to you, Charles.

(Applause.)

**CHARLES LAMANNA:** Thank you, Farah. As we think about all of these agents showing up across our applications, our teams and in more places, we're going to need to secure them. That's where we have an incredible set of capabilities. Entra manages identities so your agents behave just like users or apps do today and your entire directory. We have Purview to discover, manage, label and protect all your data as it flows throughout Copilot Teams, Copilot Studio and more. We have all this wired up into a single managed platform. To show this, I'm excited to welcome Shilpa on the stage, who will highlight all these great security features.

**SHILPA RANGANATHAN:** Thank you, Charles. Hello everyone! As developers, we're all creating agents that act autonomously, but with that power comes responsibility, especially when it comes to data security. Microsoft Purview adds built-in guardrails so that your agents don't end up accessing, exposing, or leaking sensitive data incorrectly, be it during development or at runtime. Let's have a look.

Here I'm in Copilot Studio. I'm using the same agent from Ryan. It's already set up with all of the triggers and actions, and now it's time to train this agent to work against CSX organizational data like shipments, routes, all that fancy stuff. I've already created all of these knowledge sources. Let's take a deeper look. Here I'm looking at a customer accounts table in Dataverse.

Thanks to Purview's built-in sensitivity labels, I can see that this data is labeled and protected correctly.

Now, let's look at a different knowledge source. Here I'm in SharePoint and this is where CSX is storing all of its organizational data. Here again I can see that all of this data is labeled and protected correctly, so it's safe for agents to use. With all of this set up, now let's switch cabins and take a look at how all of this works for the people at CSX using these agents.

Here, I'm in Copilot Studio. I'm using the same agent again and I'm asking it for shipments that have been delayed. My agent has gone to work and it's got me some details. Let's unpack. First and foremost, my agent is working against labeled data, so it's only going to return to me what I'm allowed to see. Second, it's also telling me very clearly in its answer the label of this content and enforcing those protections. Why is that important? Well, now I know how to handle this data as well.

But wait, this offer doesn't end here. Your agent has done even better. It's inherited the most restrictive label of all the answers for its output. What does that mean? You can take this output, create a page out of it, create a word document. Whatever you choose, the same label and corresponding protections apply there as well. All AI-generated content is automatically labeled and protected. With label inheritance, Purview adds data security right into your agent's logic. Isn't that cool?

All right. At our next arrival station, let's look at the data leak prevention scenario. Here, I'm going to ask the agent for some customer details that I know contain PII and should not be shared. For the first time on stage, I'm excited and happy to see an error message. The agent actually did the right thing. This is Purview DLP in action for you that's preventing your agent from summarizing and leaking sensitive data, all based on those same built-in labels.

From development to deployment, we saw how your agents can be secure by design, all powered by Microsoft Purview. No hacks, no heavy lifting, just smart, policy-driven enforcement built right in. You know your world's most famous agent, James Bond, he doesn't go out there without his gizmos and gadgets. Why would you let your agents go out there without Purview when it's that easy? Over to you, Charles.

**CHARLES LAMANNA:** Awesome. Thank you, Shilpa. As we look at what we're trying to do with all these security capabilities, whether it's in Copilot or Studio or Foundry, it's all about making it secure by default, so Entra and Purview protect your apps and your users, just like they've been doing for decades. Additionally, they now protect all the AI workflows and experiences that you're developing, so all those existing security models continue to work. It's one model for agents, people and apps.

With all of this being so simple and integrating to the platform, there's no reason to wait. You can go build these amazing agents today. No matter what your IT landscape looks like, there's no reason to not be ambitious and bold with how they show up. All of these AI capabilities that you've seen so far today need world-class cloud and infrastructure. To show all of that in action, Scott's going to step through some of the incredible details of how the cloud runs. I'm going to

hand it off to Scott, but just a huge thank you for sticking together for our entire demo.

(Applause.)

END