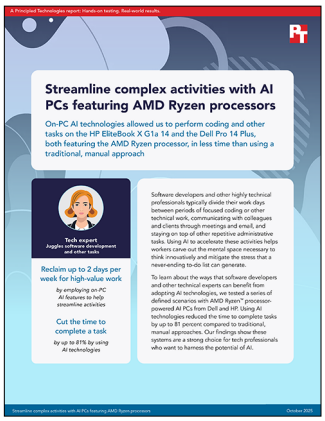

The science behind the report:

Streamline complex activities with AI PCs featuring AMD Ryzen processors

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Streamline complex activities with AI PCs featuring AMD Ryzen processors.

We concluded our hands-on testing on September 19, 2025. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on September 19, 2025 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to http://facts.pt/calculating-and-highlighting-wins. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our testing. Less time is better.

	HP EliteBook X G1a 14 AI		Dell Pro 14 Plus	
Time to complete task (mm:ss)	without AI	with AI	without AI	with AI
Summarizing an email chain	00:16:28	00:02:05	00:16:51	00:02:18
Taking notes during video conferencing	00:10:10	00:09:27	00:10:04	00:09:28
Summarizing comments on a Jira ticket	00:11:58	00:02:53	00:11:36	00:03:11
Building a predefined application	93:19:48	17:03:00	93:19:48	17:03:36
Drafting and reviewing a document	00:16:28	00:02:05	00:16:59	00:02:18
Total	**94:21:43**	**17:30:30**	**94:19:56**	**17:31:22**

Table 2: Results of our testing extrapolated over a workweek. Less time is better.

		HP EliteBook X G1a 14 AI		Dell Pro 14 Plus	
Time to complete task (mm:ss)	Iterations	without AI	with AI	without AI	with AI
Summarizing an email chain	25x weekly	06:51:48	00:52:05	07:01:15	00:57:30
Taking notes during video conferencing	10x weekly	01:41:40	01:34:27	01:40:43	01:34:37
Summarizing comments on a Jira ticket	25x weekly	04:59:18	01:12:13	04:50:00	01:19:27
Building a predefined application	1x every two months*	11:40:00	02:07:53	11:40:00	02:07:55
Drafting and reviewing a document	3x weekly*	01:09:19	00:39:10	01:04:15	00:39:19
Total (hh:mm:ss)		**26:22:06**	**06:25:48**	**26:16:13**	**06:38:47**

*For our purposes we calculated one month as four workweeks.

System configuration information

Table 3: Detailed information on the systems we tested.

System configuration information	Dell Pro 14 Plus	HP EliteBook X G1a 14
Processor		
Vendor	AMD	AMD
Model number	AMD Ryzen AI 7 PRO 350 w/ Radeon 860M	AMD Ryzen AI 9 HX PRO 375 w/ Radeon 890M
Core frequency (MHz)	2,000	2,000
Number of cores	8	12
Cache (MB)	16	24
Memory		
Amount	32	32
Type	DDR5	DDR5
Speed (MHz)	8,000	8,000
Graphics		
Vendor	AMD	AMD
Model number	AMD Radeon™ 860M Graphics	AMD Radeon 890M Graphics
Storage		
Amount	512 GB	1 TB
Type	HDD	HDD
Connectivity/expansion		
Wired internet	1 RJ45 (1 Gbps) Ethernet port	Via USB-C Adapter
Wireless internet	Wi-Fi 7 MT7925	MediaTek Wi-Fi 7 MT7925
Bluetooth	2x2, 802.11be, Bluetooth® 5.4 wireless card	Bluetooth® 5.4 wireless card
USB	1 USB 3.2 Gen 1 (5 Gbps) port with PowerShare 1 USB 3.2 Gen 1 (5 Gbps) port	1 USB Type-C® 10Gbps signaling rate 1 USB Type-A 10Gbps signaling rate
Thunderbolt	2 x USB Type-C Thunderbolt™ 4.0 with Power Delivery	Thunderbolt™ 4 with USB Type-C® 40Gbps signaling rate
Video	AMD Radeon 860M Graphics	AMD Radeon 890M Graphics
Battery		
Type	ExpressCharge™ Capable, ExpressCharge Boost Capable	HP XL-Long Life
Size	3-cell	4-cell
Rated capacity (Wh)	55	74.5

System configuration information	Dell Pro 14 Plus	HP EliteBook X G1a 14
Display		
Size (in.)	14	14
Type	LCD	LCD
Resolution	1,920 x 1,200	1,920 x 1,200
Touchscreen	Yes	Yes
Operating system		
Vendor	Microsoft	Microsoft
Name	Microsoft Windows 11 Enterprise	Microsoft Windows 11 Enterprise
Build number or version	10.0.26100 (Build 26100)	10.0.26100 (Build 26100)
BIOS		
BIOS name and version	Dell Inc. 2.1.5	HP X90 Ver. 01.02.03
Dimensions		
Height (in.)	0.80	0.52
Width (in.)	12.30	12.29
Depth (in.)	8.80	8.45
Weight (lb.)	3.51	3.3

How we tested

Summarizing an email thread

We conducted a comparative evaluation of Microsoft Copilot's AI-powered email summarization against traditional manual email reading. This test employed realistic email threads to simulate authentic workplace scenarios, enabling an accurate assessment of productivity differences in information processing.

AI-enhanced workflow

1. Open a web browser.
2. Go to https://outlook.office.com.
3. Type your email address.
4. Press Enter or click Next.
5. Type your password.
6. Press Enter or click Sign in.
7. Click on the first email in the list (topmost message).
8. Wait for the email to open and display its contents.
9. Prepare the stopwatch.
10. Simultaneously start the stopwatch and click the Copilot button.
11. In the Message Copilot field, type `Summarize the thread titled Urgent: Website Updates for MexiCat Pet Feeder Launch` and press Enter.
12. From the Copilot menu, click Summarize.
13. Wait for Copilot to generate and display the email summary.
14. When the summary is completely generated, read through it once and stop the stopwatch on completion.
15. Record the result.
16. Repeat steps 1 through 15 two more times.

Traditional workflow

1. Open a web browser.
2. Go to https://outlook.office.com.
3. Type your email address.
4. Press Enter or click Next.
5. Type your password.
6. Press Enter or click Sign in.
7. Click on the first email in the list (topmost message).
8. Wait for the email to open and display its contents.
9. Prepare the stopwatch.
10. Simultaneously start the stopwatch and read through the thread.
11. After you have finished reading through the thread, open Word and write a summary.
12. When the summary is complete, stop the stopwatch.
13. Record the result.
14. Repeat steps 1 through 13 two more times.

Video Conferencing with Teams AI Note-Taking

AI-enhanced workflow

1. Launch Camo Studio by running `C:\Program Files (x86)\Camo Studio\camostudio.exe`
2. In Camo Studio, configure it to use the video file AMDClip.mp4 from the resources folder.
3. Wait 15 seconds for Camo Studio to fully load.
4. Minimize the Camo Studio window.
5. Open the Microsoft Teams application.
6. Click Calendar view.
7. Click Meet now.
8. Click Start meeting.
9. Configure meeting settings to use Camo Studio as the camera source.

10. Join the meeting with video enabled.
11. Launch Windows Media Player Legacy by running `C:\Program Files (x86)\Windows Media Player\wmplayer.exe`
12. In Windows Media Player, open and play the audio file AMDClipAudio.mp3 from the resources folder.
13. Minimize the Windows Media Player window.
14. Let the meeting run for 5 minutes (300 seconds).
15. After 5 minutes, close Camo Studio completely.
16. Close Windows Media Player.
17. In Teams, leave the meeting.
18. Wait 1 minute for Copilot AI Recap to generate.
19. Open the Chat pane.
20. Navigate to the latest meeting.
21. Open Recap.
22. Maximize the recap window.
23. Open the Copilot pane.
24. Enter the following prompt:

```
Were there any action items to take note of?
```

25. Navigate to Transcript save options.
26. Click Save Transcript to Downloads.
27. Close Microsoft Teams.

Traditional workflow

1. Open the Microsoft Teams application.
2. Navigate to the Calendar view.
3. Click Meet now.
4. Click Start meeting.
5. Configure meeting settings to use Camo Studio as the camera source.
6. Join the meeting with video enabled.
7. Launch Windows Media Player Legacy by running `C:\Program Files (x86)\Windows Media Player\wmplayer.exe`
8. In Windows Media Player, open and play the audio file AMDClipAudio.mp3 from the resources folder.
9. Minimize the Windows Media Player window.
10. Launch Microsoft Word.
11. Wait for Word to fully load (about 30 seconds).
12. Press Enter to select Blank Document from the welcome screen.
13. Wait an additional 5 seconds before beginning note-taking.
14. Minimize the main Teams window while keeping the meeting window active.
15. Keep Word in the foreground and begin taking meeting notes.
16. Type the document header `MexiCat Project Meeting Notes` with the current date.
17. Press Enter twice to create space for notes.
18. During the meeting, take timestamped notes about key discussion points including project vision, product features, marketing strategy, and technical requirements.
19. Type notes in a rushed, informal style with occasional typos and abbreviations to simulate real-time note-taking.
20. Continue taking notes for the full duration of the meeting (approximately 5 minutes).
21. When the meeting ends, minimize Word and wait for cleanup phase.
22. After the meeting concludes, restore Word to the foreground.
23. Go to the end of the document using Ctrl+End.
24. Add a separator line and create a “CLEANED UP MEETING NOTES” section.
25. Add a professional meeting overview summarizing the key points discussed.
26. Create organized sections for “KEY DISCUSSION POINTS,” “ACTION ITEMS,” and “NEXT STEPS.”
27. Include attendee information and meeting date in the cleaned-up section.
28. Save the document.
29. Close Word.
30. Return to the Teams application window and leave the meeting.
31. Check for and dismiss any call quality feedback prompts that appear.
32. If a feedback dialog appears, click Dismiss or Close to dismiss it.
33. Stop timer.

Summarizing the comments in a Jira ticket

AI-enhanced workflow

1. Open a browser and go to following link: https://radteam-lab.atlassian.net/jira/software/projects/KAN/list?atlOrigin=eyJpIjoiOTZmZWJkYTJmNTI1NDAxN2IzOWQzNjI5M2Y4YzgzYTliLCJwIjoiaiJ9
2. Enter your username and password and click Log in.
3. Ensure User is logged into JIRA with appropriate permissions.
4. Ensure Atlassian Intelligence summarization is enabled for the project/instance.
5. Under the Projects there is MexiCat – navigate to List view, expand the epic Picture 1, Picture "New Product Launch: MexiPetFeeder to MexiCat Line."
6. Prepare the stopwatch.
7. Simultaneously start the stopwatch and click JIRA bug ticket KAN-36 "BETA TESTER FEEDBACK: Catastrophic UI Input - Malicious Firmware Update Trigger & Device Bricking."
8. The ticket should have a pop-up expanding all details.
9. Click Picture 1, the picture in upper right.
10. Click Summarize comments.
11. Under the Ticket's Activity, an inner window will appear showing Picture 1. Atlassian Intelligence's comment summary will appear in this inner window.
12. Read through the summary to verify the summary accurately reflects the content of the various comments.
13. Copy the summary and past it in an email.
14. Stop the stopwatch and record the time.
15. Repeat steps 1 through 14 two more times with different users.

Traditional workflow

1. Open a browser and go to following link: https://radteam-lab.atlassian.net/jira/software/projects/KAN/list?atlOrigin=eyJpIjoiOTZmZWJkYTJmNTI1NDAxN2IzOWQzNjI5M2Y4YzgzYTliLCJwIjoiaiJ9.
2. Enter your username and password and click Log in.
3. Ensure User is logged into JIRA with appropriate permissions.
4. Ensure Atlassian Intelligence summarization is enabled for the project/instance.
5. Under the Projects there is MexiCat – navigate to List view, expand the epic Picture 1, Picture "New Product Launch: MexiPetFeeder to MexiCat Line."
6. Simultaneously start the stopwatch and click JIRA bug ticket KAN-36 "BETA TESTER FEEDBACK: Catastrophic UI Input - Malicious Firmware Update Trigger & Device Bricking."
7. The ticket should have a pop-up expanding all details.
8. Read through the entire ticket and comments.
9. Write a summary of ticket and comments in an email.
10. Stop the stopwatch when the summary is complete, and record the time.
11. Repeat steps 1 through 10 two more times with different users.

Building a predetermined application (vibe coding) with AI assistance

Our AI-enhanced workflow used a local LLM hosted by Lemonade Server, which we accessed directly from Continue.dev in VS Code. This provided AI-assisted code generation and error checking without switching applications. We completed the workflow three times to determine the median completion time.

1. Launch Lemonade Server, and load Qwen3-Coder-30B-A3B-Instruct-GGUF (for its suitability for coding).
2. Open VS Code, and confirm Continue.dev is installed and signed in.
3. Configure Continue.dev to point to your local Lemonade endpoint (default: http://localhost:8000/api/v1).
4. Prompt the AI assistant in VS Code to create the initial web application.
5. Review the AI-generated code directly in the VS Code editor with the Problems panel for quick error detection.
6. Run and test the final application with Node.js, and curl to verify endpoints. (Windows 11 contains Curl by default. You can also download and install from https://curl.se/windows/.)

Installing tools

Installing Lemonade Server

1. Open a browser.
2. Navigate to https://lemonade-server.ai/.
3. Download Windows 11.
4. Save Lemonade_Server_Installer.exe file locally.
5. Run Lemonade_Server_Installer.exe.
6. At the Welcome screen, click Next.
7. At the Choose Components screen, accept defaults, and click Next.
8. At the Select Installation Directory, accept defaults, and click Install.

Installing VS Code

1. Open a browser.
2. Navigate to https://code.visualstudio.com/download.
3. Click the Windows 10, 11 button.
4. Save VSCodeUserSetup-x64-1.104.3.exe locally.
5. Run VSCodeUserSetup-x64-1.104.3.exe.
6. At the License Agreement screen, select I accept the agreement, and click Next.
7. At the Select Additional Tasks screen, accept defaults, and click Next.
8. At the Ready to Install screen, click Install.

Installing Continue.dev

1. Open VS Code.
2. On the left sidebar, click Extensions.
3. Type `Continue` in the search bar, and on the Continue - open-source AI code agent extension, click Install.

Installing Node.js

1. Open a browser.
2. Navigate to https://nodejs.org/en/download.
3. Click the Windows Installer (.msi) button.
4. Save node-v22.20.0-x64.msi locally.
5. Run node-v22.20.0-x64.msi.
6. At the Welcome screen, click Next.
7. At the License Agreement screen, click I accept the terms in the license agreement, and click Next.
8. At the Destination Folder screen, accept defaults, and click Next.
9. At the Custom Setup screen, accept defaults, and click Next.
10. At the Tools for Native Modules screen, accept defaults, and click Next.
11. At the Ready to Install Node.js screen, click Install.

Preparing to test

Setting up Lemonade Server

1. Launch Lemonade Server.
2. Lemonade Server will run in the system tray. Right-click the lemon, and select Model Manager.
3. On the left sidebar, click Coding.
4. Download the Qwen3-Coder-30B-A3B-Instruct-GGUF model.
5. Allow the model to fully download.
6. Setting up Continue.dev
7. Open VS Code.
8. On the left side of VS Code, click Continue in the Activity Bar to open the Continue sidebar.
9. In the Continue sidebar, click the agent selector drop-down menu above the chat dialog box, hover over Local Agent, and click the gear to open your config file (config.yaml). This will add a Local Agent pointing to Lemonade.

10. Paste this minimal model block into config.yaml:

```
name: Lemonade Local
version: "1"
schema: v1
models:
name: Qwen3 Coder 30B (Lemonade)
provider: openai
model: Qwen3-Coder-30B-A3B-Instruct-GGUF
apiBase: "http://localhost:8000/api/v1"
apiKey: "unused"
roles: [chat, edit, apply, summarize]
defaultCompletionOptions:
temperature: 0.2
maxTokens: 1024
```

11. Save the file. Click Reload config in the Continue sidebar. Select Qwen3 Coder 30B (Lemonade) from the agent drop-down menu.
12. To before a quick smoke test, in the Continue chat box, type a reply with the word `READY` only. You should receive the response READY.

Preparing the local LLM server

1. Launch Lemonade Server.
2. Lemonade Server will run in the system tray. Right-click the lemon, and select Model Manager.
3. Prepare the stopwatch.
4. Simultaneously start the stopwatch and, from the drop-down menu, select Qwen3-Coder-30B-A3B-Instruct-GGUF. When the model has loaded, the model name will turn green in the upper-right corner of the web UI.

Configuring Continue.dev in VS Code

1. Open VS Code.
2. To open the Continue sidebar, click the Continue icon in the Activity Bar on the left side of VS Code.
3. On the Continue sidebar, click the + to start a new session.
4. Ensure Qwen3 Code 30B (Lemonade) is selected in the model drop-down menu.

Creating the initial web application (HTTP server)

1. In the Continue.dev chat panel in VS Code, paste the following prompt:

```
Create an HTTP server with Javascript on port 3000 with these exact routes:
- GET /api/users - return users array
- POST /api/users - create user from JSON body
Use in-memory storage with default user 'mexicat', include CORS, JSON parsing, and basic error
handling. Single file only.
```

2. Wait for the model to finish generating its response. Copy the created js contents into a new file in VS Code.
3. With the new file tab open, go to File → Save As…, choose the folder in which you're working, and name the file `http_server.js`. Click Save.

Verifying code is error-free

Check the Problems panel at the bottom of VS Code (Ctrl+Shift+M) and verify that there are no issues.

Testing the application

1. Open the Terminal panel at the bottom of VS Code.
2. Navigate to the folder containing the http_server.js file.
3. To start the server, run `node http_server.js`.
4. Verify that the console outputs Server running on port 3000.
5. Open a new Terminal tab (Ctrl+Shift+`) to keep the server running.
6. To test the GET endpoint, run `curl.exe http://localhost:3000/api/users`. The HTTP server should respond with information about the created user. Responses won't always be the same given the vagueness of the creation prompt.
7. Upon verification of the endpoint, stop the stopwatch and record the time.

Drafting and reviewing a document

Note: In this scenario, we measured how long it took to launch LM Studio, load a specified AI model, generate a document draft using an attached reference document, transfer the draft to Microsoft Word, review and finalize the content, and save the completed document. Our testing used a sample document that already contained a structure for the document. While most technical users may not routinely draft such documents, they are likely to have to produce summary-style documents—short, structured, and tailored to specific audiences—such as release notes, update briefs, or internal rollout guides. For more information on our test methodology, please contact info@principledtechnologies.com.

AI-enhanced workflow

1. Simultaneously start the timer and launch LM Studio.
2. Start a new conversation.
3. Load the designated model.
   - Wait for the model to fully load before proceeding.
4. Submit the prompt as specified.
   - Include reference material if needed.
   - Request a first draft of a product announcement document.
5. Wait for the model to complete its response.
6. Record performance metrics.
   - Tokens per second
   - Total tokens generated
   - Word count of the response
7. Open a word processor.
8. Copy and paste the generated draft.
   - Record the word count after insertion.
9. Review the content for accuracy and alignment.
   - Ensure the document meets the intended goal.
10. Fill in any placeholder details.
11. Save the completed document.
12. Stop the timer.
13. Record the total time.
14. Repeat the process two more times.

Traditional workflow

1. Simultaneously start the timer and launch Microsoft Word
2. Open the reference file for product specifications.
3. Create the first draft of the document using placeholders for key details.
4. Perform a brief first draft review.
   - Ensure the writing is largely error-free and structurally sound.
5. Fill in placeholder details with the specified information.
6. Save the completed document.
7. Stop the timer once the document is saved.
8. Record the total time.
9. Repeat steps 1 through 8 two more times.

Read the report ▶

This project was commissioned by AMD.

Principled Technologies®

Facts matter.®