WEEK 2

Advanced Prompts & Your First Agent

AI Productivity Training for Tesla Operations Leaders

240 minutes | Hands-On Workshop | 81% hands-on time

Today's Agenda

0:00	Homework Review & Wins	10 min
0:10	Advanced Prompting: CoT, Few-Shot, Persona	5 min
0:15	**Exercise 1:** Data Analysis Sprint	30 min
0:45	**Exercise 2:** Template Speed Build	10 min
0:55	*BREAK*	10 min
1:05	What Is an AI Agent?	5 min
1:10	**Exercise 3:** VS Code + Copilot Setup	20 min
1:30	Copilot Modes: Chat vs Edit vs Agent	5 min
1:35	**Exercise 4:** Your First Agent Mode Task	25 min
2:00	*BREAK*	10 min
2:10	MCP: Giving Agents Skills	5 min
2:15	**Exercise 5:** Install First MCP Skill	25 min
2:40	Reading Agent Plans & Git Basics	5 min
2:45	**Exercise 6:** Agent Builds a Real Tool	30 min
3:15	**Exercise 7:** Debug & Redirect an Agent	25 min
3:40	Takeaways + Homework	20 min

01

SECTION 1

Homework Review & Wins

10 minutes

Peer Sharing

SECTION 1 — HOMEWORK REVIEW

Quick Round: Share Your Wins

1

What was your **best prompt** this week? What made it work?

2

What was your **worst prompt?** What went wrong?

3

How much **time did you save?**
Target: 15-20 min/day

4

If you saved **30+ min/day** — tell us how!

02

SECTION 2

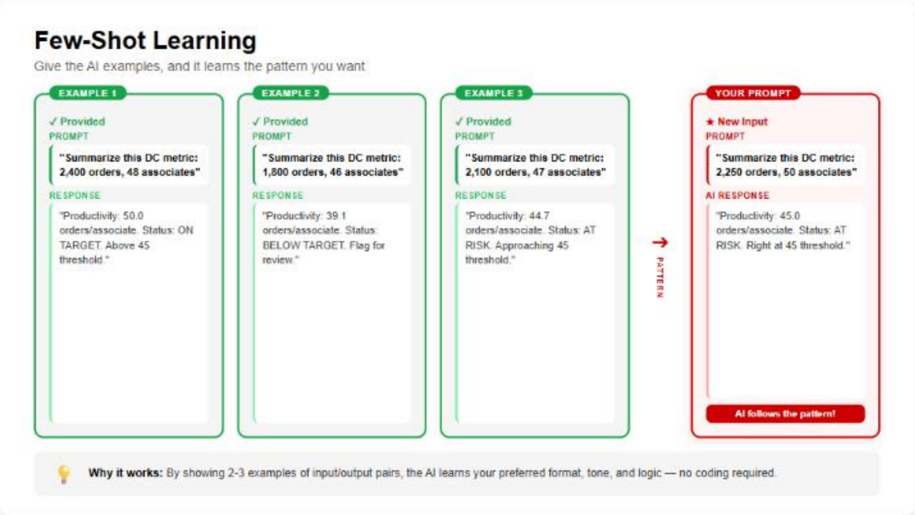
Advanced Prompting Techniques

5 minutes

Concept Intro

~90 seconds per technique

Three Techniques That Level Up Your Prompts



Few-Shot Learning

Give the AI examples, and it learns the pattern you want

EXAMPLE 1

✓ Provided

PROMPT

"Summarize this DC metric: 2,400 orders, 48 associates"

RESPONSE

"Productivity: 50.0 orders/associate. Status: ON TARGET. Above 45 threshold."

EXAMPLE 2

✓ Provided

PROMPT

"Summarize this DC metric: 1,800 orders, 46 associates"

RESPONSE

"Productivity: 39.1 orders/associate. Status: BELOW TARGET. Flag for review."

EXAMPLE 3

✓ Provided

PROMPT

"Summarize this DC metric: 2,100 orders, 47 associates"

RESPONSE

"Productivity: 44.7 orders/associate. Status: AT RISK. Approaching 45 threshold."

→ PATTERN

YOUR PROMPT

★ New Input

PROMPT

"Summarize this DC metric: 2,250 orders, 50 associates"

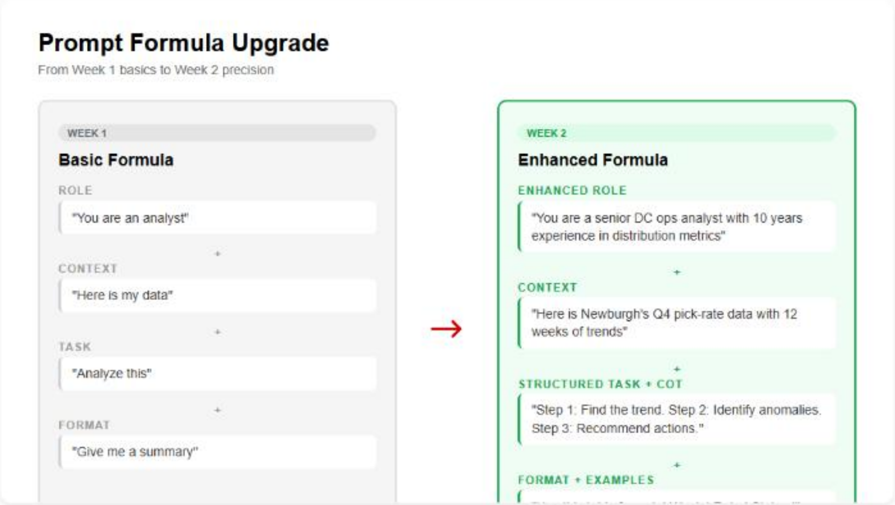
AI RESPONSE

"Productivity: 45.0 orders/associate. Status: AT RISK. Right at 45 threshold."

AI follows the pattern!

💡 **Why it works:** By showing 2-3 examples of input/output pairs, the AI learns your preferred format, tone, and logic — no coding required.

The Formula Upgrade



Prompt Formula Upgrade

From Week 1 basics to Week 2 precision

WEEK 1

Basic Formula

ROLE

"You are an analyst"

+

CONTEXT

"Here is my data"

+

TASK

"Analyze this"

+

FORMAT

"Give me a summary"

→

WEEK 2

Enhanced Formula

ENHANCED ROLE

"You are a senior DC ops analyst with 10 years experience in distribution metrics"

+

CONTEXT

"Here is Newburgh's Q4 pick-rate data with 12 weeks of trends"

+

STRUCTURED TASK + COT

"Step 1: Find the trend. Step 2: Identify anomalies. Step 3: Recommend actions."

+

FORMAT + EXAMPLES

03

SECTION

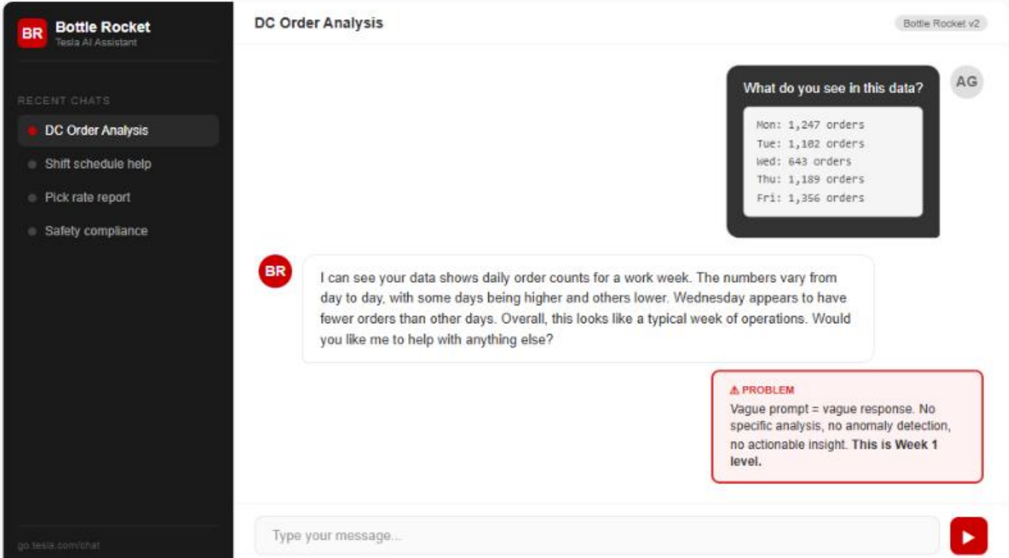
Exercise 1: CoT + Data Analysis Sprint

30 MIN HANDS-ON IN **BOTTLE ROCKET**

- Part A: See the Difference (15 min)
- Part B: Your Own Data (15 min)

EXERCISE 1 — PART A, STEP 1

The Basic Prompt



What do you see in this data?

```
Mon: 1,247 orders
Tue: 1,182 orders
Wed: 643 orders
Thu: 1,189 orders
Fri: 1,356 orders
```

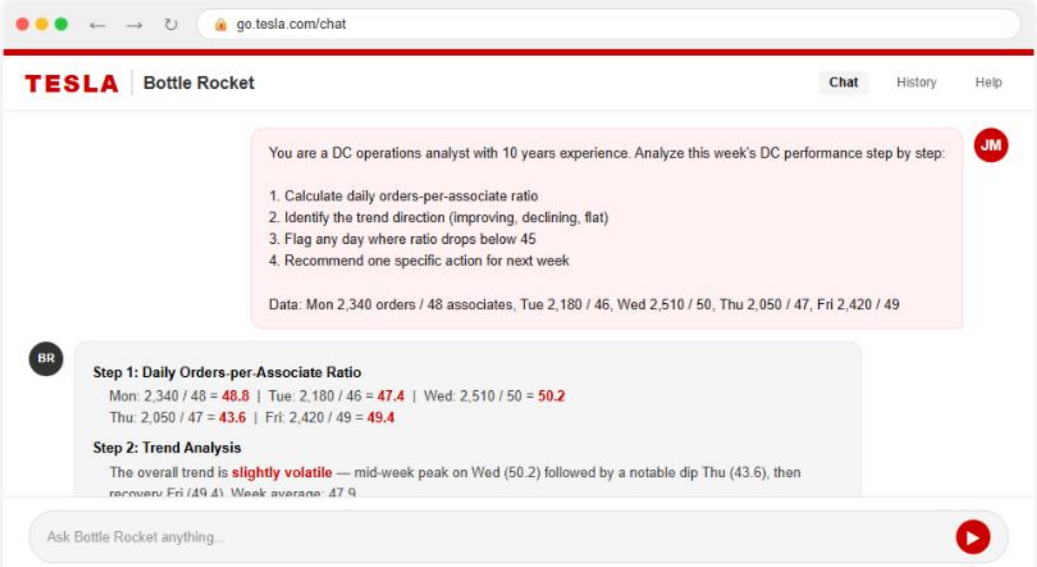
I can see your data shows daily order counts for a work week. The numbers vary from day to day, with some days being higher and others lower. Wednesday appears to have fewer orders than other days. Overall, this looks like a typical week of operations. Would you like me to help with anything else?

⚠ PROBLEM

Vague prompt = vague response. No specific analysis, no anomaly detection, no actionable insight. **This is Week 1 level.**

EXERCISE 1 — PART A, STEP 2

The CoT Prompt



You are a DC operations analyst with 10 years experience. Analyze this week's DC performance step by step:

1. Calculate daily orders-per-associate ratio
2. Identify the trend direction (improving, declining, flat)
3. Flag any day where ratio drops below 45
4. Recommend one specific action for next week

Data: Mon 2,340 orders / 48 associates, Tue 2,180 / 46, Wed 2,510 / 50, Thu 2,050 / 47, Fri 2,420 / 49

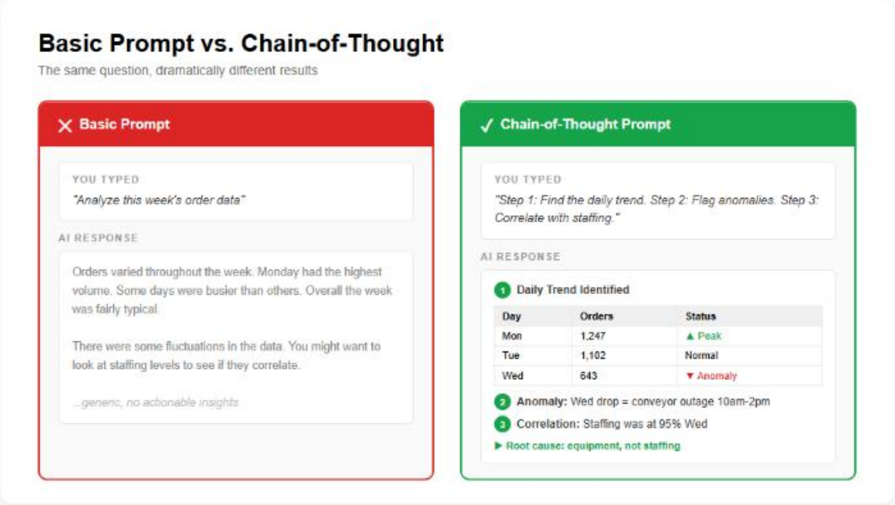
Step 1: Daily Orders-per-Associate Ratio

Mon: 2,340 / 48 = **48.8** | Tue: 2,180 / 46 = **47.4** | Wed: 2,510 / 50 = **50.2**
Thu: 2,050 / 47 = **43.6** | Fri: 2,420 / 49 = **49.4**

Step 2: Trend Analysis

The overall trend is **slightly volatile** — mid-week peak on Wed (50.2) followed by a notable dip Thu (43.6), then recovery Fri (49.4). Week average: 47.9

What to Look For: Basic vs CoT



Basic Prompt vs. Chain-of-Thought

The same question, dramatically different results

✕ Basic Prompt

YOU TYPED

"Analyze this week's order data"

AI RESPONSE

Orders varied throughout the week. Monday had the highest volume. Some days were busier than others. Overall the week was fairly typical.

There were some fluctuations in the data. You might want to look at staffing levels to see if they correlate.

...generic, no actionable insights

✓ Chain-of-Thought Prompt

YOU TYPED

"Step 1: Find the daily trend. Step 2: Flag anomalies. Step 3: Correlate with staffing."

AI RESPONSE

① Daily Trend Identified

Day	Orders	Status
Mon	1,247	▲ Peak
Tue	1,102	Normal
Wed	643	▼ Anomaly

② **Anomaly:** Wed drop = conveyor outage 10am-2pm

③ Correlation: Staffing was at 95% Wed

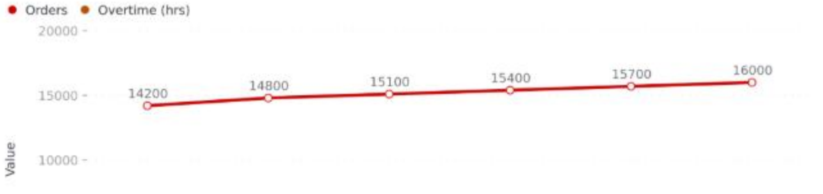
▶ **Root cause: equipment, not staffing**

EXERCISE 1 — PART B

Your Own 6-Week Data Analysis

TYPE THIS IN BOTTLE ROCKET

```
"You are a senior operations analyst specializing in DC performance trends."
"Analyze this 6-week data step by step: 1) Identify trend for each metric
2) Find correlations 3) Flag outlier weeks 4) Provide 3 action items
ranked by impact"
```



6-Week DC Performance Trends

Exercise 1: Success Checklist

- ✓ You see a between basic and CoT prompts
- ✓ Your Part B analysis identifies at least
- ✓ Your action items include , not vague suggestions

COMMON FIX

If AI gives generic recommendations, add: *"Include specific numbers and timelines"*

04

SECTION

10 minutes

Exercise 2:
Template Speed Build

● 5 min build ● 3 min test ● 2 min save

EXERCISE 2

Upgrade One Template

TYPE THIS IN BOTTLE ROCKET — ADAPT TO YOUR TASK

```
"You are a [your role] with [X] years of experience
managing [your specialty]."

"I need a reusable template for [your weekly task].
Build it with:"

"1. Use [PLACEHOLDERS] for data that changes each week"
"2. Include step-by-step analysis instructions (Chain-of-Thought)"
"3. Include one example of the expected output format"

"The template should produce output ready to send to
[your audience] without editing."
```

Replace all **[YELLOW PLACEHOLDERS]** with your real role, task, and audience before running

Template Success Check

- ✓ Uses at least one Week 2 technique (CoT, Few-Shot, or enhanced persona)
- ✓ Has [PLACEHOLDERS] for variable data
- ✓ Tested and produces work-ready output
- ✓ Save it to your prompt library — you will use it this week

Take 10 minutes. When we come back, everything changes.

BREAK

10 minutes

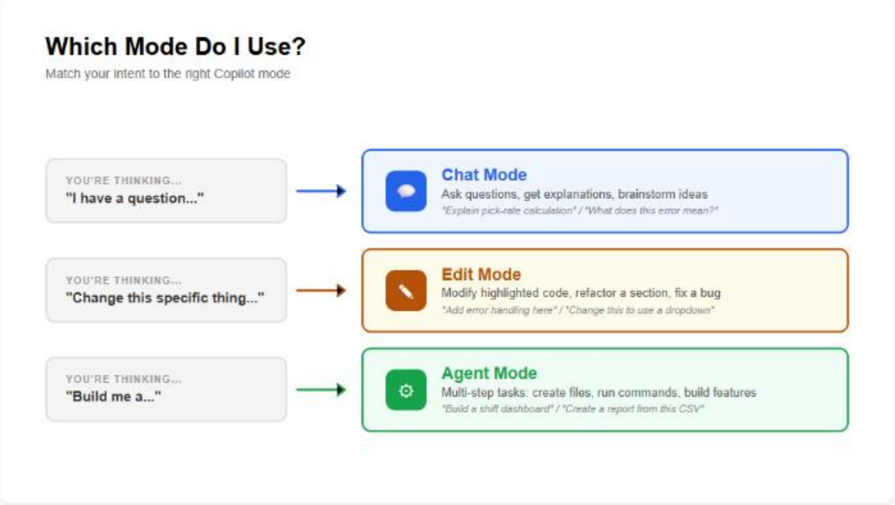
Back at 1:05

05

What Is an AI Agent?

5 minutes Concept Intro

Chat vs Agent — The Core Difference



Which Mode Do I Use?

Match your intent to the right Copilot mode

YOU'RE THINKING...
"I have a question..."

Chat Mode
Ask questions, get explanations, brainstorm ideas
"Explain pick-rate calculation" / "What does this error mean?"

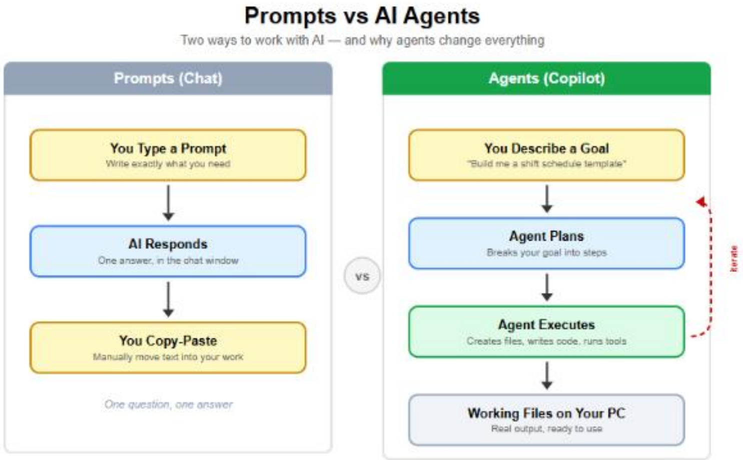
YOU'RE THINKING...
"Change this specific thing..."

Edit Mode
Modify highlighted code, refactor a section, fix a bug
"Add error handling here" / "Change this to use a dropdown"

YOU'RE THINKING...
"Build me a..."

Agent Mode
Multi-step tasks: create files, run commands, build features
"Build a shift dashboard" / "Create a report from this CSV"

Prompts vs AI Agents



Prompts vs AI Agents

Two ways to work with AI — and why agents change everything

Prompts (Chat)

You Type a Prompt
Write exactly what you need

AI Responds
One answer, in the chat window

You Copy-Paste
Manually move text into your work

One question, one answer

vs

Agents (Copilot)

You Describe a Goal
"Build me a shift schedule template"

Agent Plans
Breaks your goal into steps

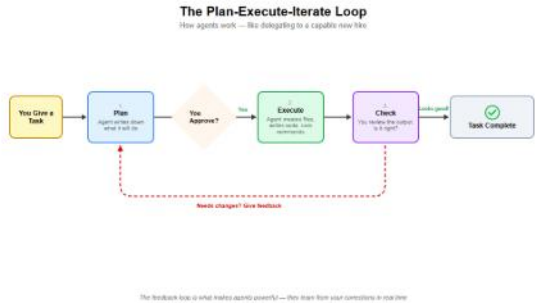
Agent Executes
Creates files, writes code, runs tools

iterate

Working Files on Your PC
Real output, ready to use

Prompts: one response. Agents: plan, execute, iterate until done.

The Plan-Execute-Iterate Loop


The Plan-Execute-Iterate Loop
How agents work — like delegating to a capable new hire
You Give a Task
1
Plan
Agent writes down what it will do
You Approve?
Yes
2
Execute
Agent creates files, writes code, runs commands
3
Check
You review the output. Is it right?
Looks good!
Task Complete
Needs changes? Give feedback
The feedback loop is what makes agents powerful — they learn from your corrections in real time

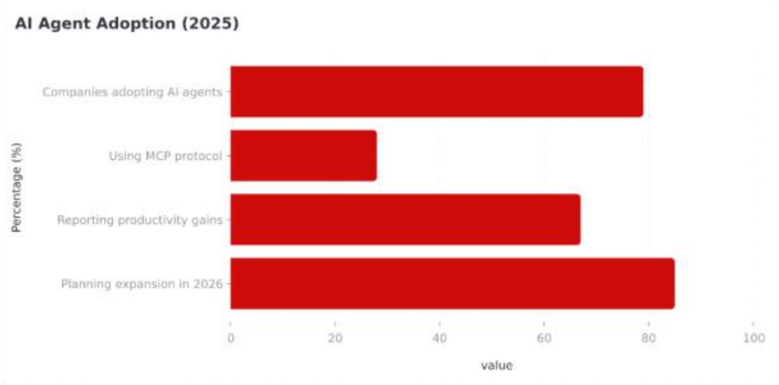
Think of delegating to a capable new hire:

1. **Plan** — Agent creates a step-by-step plan
2. **Execute** — Agent carries out each step
3. **Check** — You review the output and give feedback

Loop continues until the task is done

"The same way you delegate to a new hire: plan, execute, check, iterate."

Why Agents Matter for Your Work



06

Exercise 3:
VS Code + Copilot Setup

20 minutes

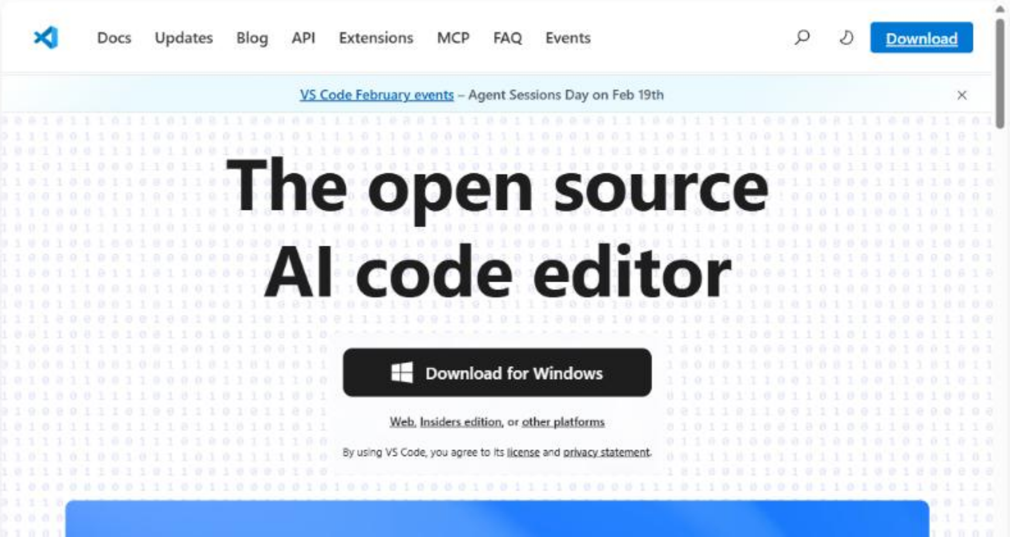
Guided Setup

POLICY REMINDER — Read This Aloud

- GitHub Copilot is **CONDITIONALLY APPROVED** at Tesla
- **NO Tesla data in Copilot — ever. No metrics, employee names, vendor details, internal processes.**
- Use **generic or fictional data** for all Copilot exercises today
- **Bottle Rocket** (go.tesla.com/chat) remains the tool for Tesla data
- Reference: **go.tesla.com/aitools**

EXERCISE 3 — STEP 1

Step 1: Download VS Code



VS Code February events – Agent Sessions Day on Feb 19th

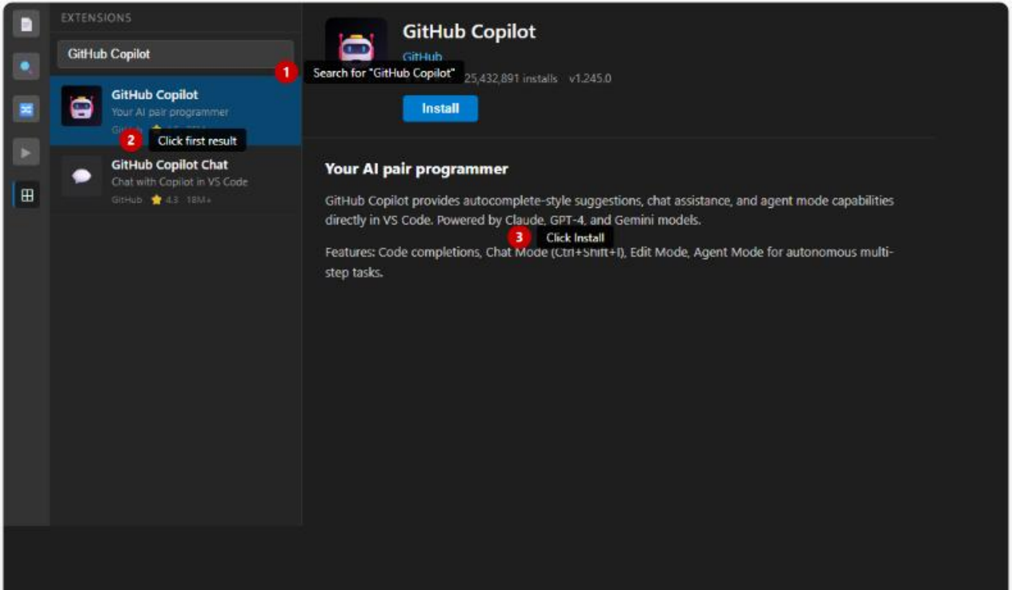
The open source AI code editor

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

EXERCISE 3 — STEP 2

Install the Copilot Extension



GitHub Copilot

GitHub Copilot
Your AI pair programmer

GitHub Copilot Chat
Chat with Copilot in VS Code
GitHub ⭐ 4.3 18M+

1 Search for "GitHub Copilot"

2 Click first result

3 Click Install

GitHub Copilot

GitHub

25,432,891 installs v1.245.0

Your AI pair programmer

GitHub Copilot provides autocomplete-style suggestions, chat assistance, and agent mode capabilities directly in VS Code. Powered by Claude, GPT-4, and Gemini models.

Features: Code completions, Chat Mode (Ctrl+Shift+I), Edit Mode, Agent Mode for autonomous multi-step tasks.

EXERCISE 3 — STEP 3

Sign In with GitHub

DO THIS

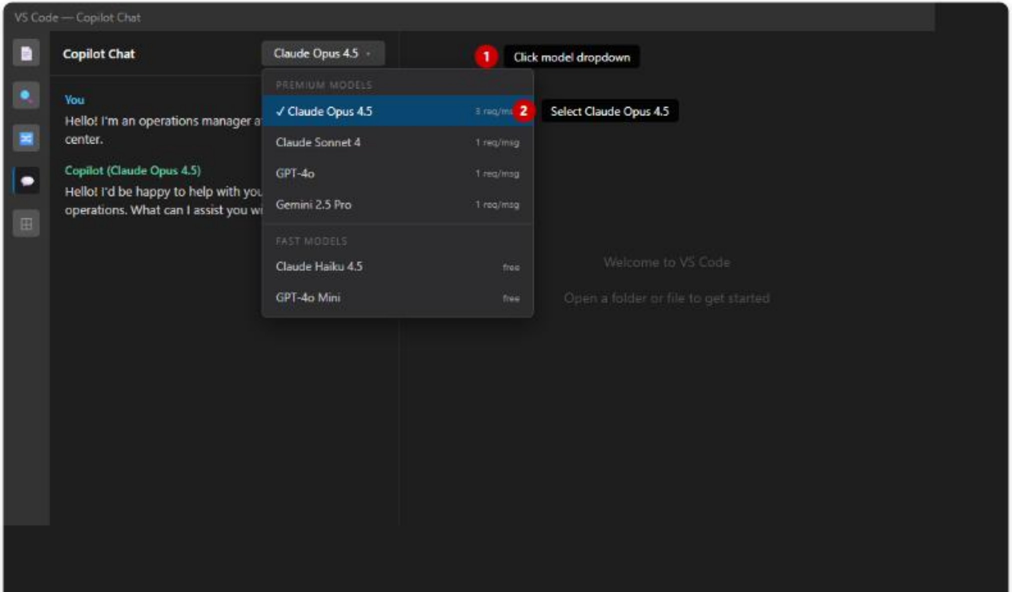
1. Click the **Copilot icon** in the left sidebar
2. Click **“Sign in to GitHub”**
3. Browser opens — enter your **GitHub username and password**
4. Click **“Authorize”** when asked
5. Switch back to **VS Code**

✓ **You should see:** Your GitHub username in the bottom-left corner

No GitHub account? Go to github.com and sign up with your personal email.

EXERCISE 3 — STEP 4

Select Claude Opus 4.5 Model



1 Click model dropdown

2 Select Claude Opus 4.5

You

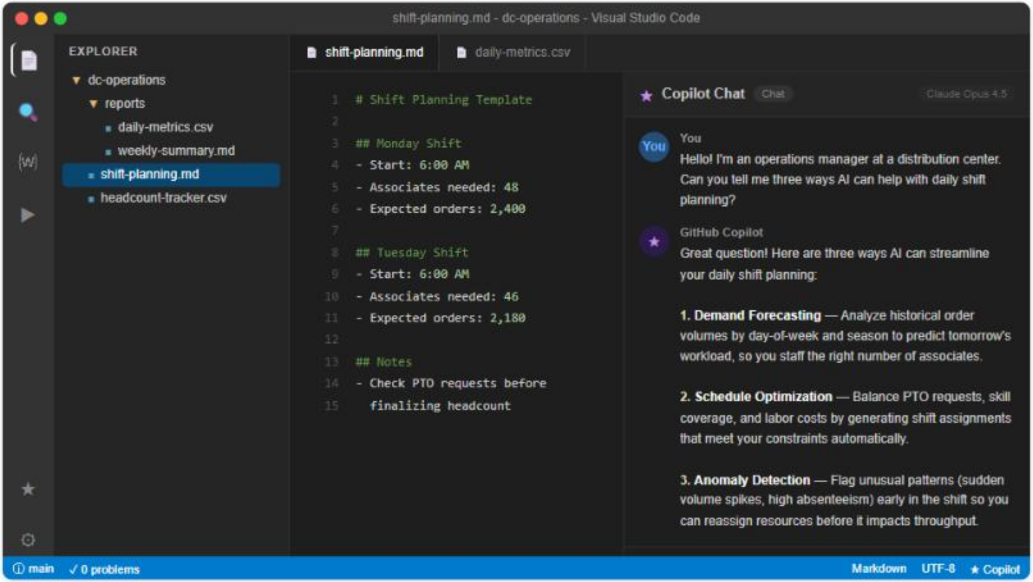
Hello! I'm an operations manager a
center.

Copilot (Claude Opus 4.5)

Hello! I'd be happy to help with you
operations. What can I assist you wi

EXERCISE 3 — STEP 5

Verify — Send Your First Message



```
# Shift Planning Template

## Monday Shift
- Start: 6:00 AM
- Associates needed: 48
- Expected orders: 2,400

## Tuesday Shift
- Start: 6:00 AM
- Associates needed: 46
- Expected orders: 2,180

## Notes
- Check PTO requests before
  finalizing headcount
```

You

Hello! I'm an operations manager at a distribution center. Can you tell me three ways AI can help with daily shift planning?

GitHub Copilot

Great question! Here are three ways AI can streamline your daily shift planning:

1. **Demand Forecasting** — Analyze historical order volumes by day-of-week and season to predict tomorrow's workload, so you staff the right number of associates.

2. **Schedule Optimization** — Balance PTO requests, skill coverage, and labor costs by generating shift assignments that meet your constraints automatically.

3. **Anomaly Detection** — Flag unusual patterns (sudden volume spikes, high absenteeism) early in the shift so you can reassign resources before it impacts throughput.

Troubleshooting Common Issues

“Sign in” keeps looping → Clear browser cookies, try incognito window

Extension not showing → Close and reopen VS Code

“You don’t have access to Copilot” → Verify GitHub subscription with facilitator

Chat not responding → Check internet, try a different model

Model dropdown empty → Update VS Code (Help > Check for Updates)

07

SECTION 7

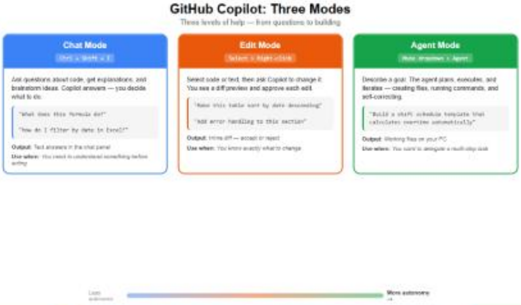
Copilot Modes: Chat vs Edit vs Agent

5 minutes

Concept Intro

Three levels of help

GitHub Copilot: Three Modes



GitHub Copilot: Three Modes

Three levels of help — from questions to building

Chat Mode

Ctrl + Shift + I

Ask questions about code, get explanations, and brainstorm ideas. Copilot answers — you decide what to do.

"What does this formula do?"

"How do I filter by date in Excel?"

Output: Text answers in the chat panel

Use when: You need to understand something before acting

Edit Mode

Select > Right-click

Select code or text, then ask Copilot to change it. You see a diff preview and approve each edit.

"Make this table sort by date descending"

"Add error handling to this section"

Output: Inline diff — accept or reject

Use when: You know exactly what to change

Agent Mode

Mode dropdown > Agent

Describe a goal. The agent plans, executes, and iterates — creating files, running commands, and self-correcting.

"Build a shift schedule template that calculates overtime automatically"

Output: Working files on your PC

Use when: You want to delegate a multi-step task

Chat Mode

`Ctrl+Shift+I`

Ask questions, get text answers. Like Bottle Rocket.

Edit Mode

`Select > Right-click`

Point at something, ask for targeted changes.

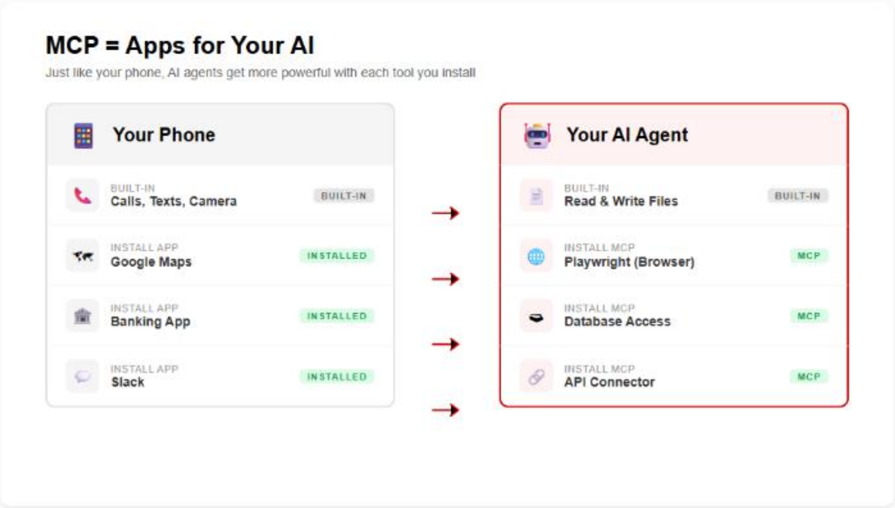
Agent Mode

`Mode dropdown > Agent`

Describe a goal, agent plans and builds.

Autonomy increases left to right: Chat < Edit < Agent

Which Mode to Use When



MCP = Apps for Your AI

Just like your phone, AI agents get more powerful with each tool you install

Your Phone			Your AI Agent	
BUILT-IN **Calls, Texts, Camera**	BUILT-IN	→	BUILT-IN **Read & Write Files**	BUILT-IN
INSTALL APP **Google Maps**	INSTALLED	→	INSTALL MCP **Playwright (Browser)**	MCP
INSTALL APP **Banking App**	INSTALLED	→	INSTALL MCP **Database Access**	MCP
INSTALL APP **Slack**	INSTALLED	→	INSTALL MCP **API Connector**	MCP

Keyboard Shortcuts to Remember

Chat Mode	`Ctrl+Shift+I`	*or click chat icon in sidebar*
Edit Mode	`Select text > Right-click > Copilot > Edit`	
Agent Mode	`Chat panel > Mode dropdown > "Agent"`	

Today we focus on Agent Mode — the most powerful option

08

SECTION 8

Exercise 4: Your First Agent Mode Task

25 minutes

Setup: **2 min**

Run agent: **10 min**

Review: **5 min**

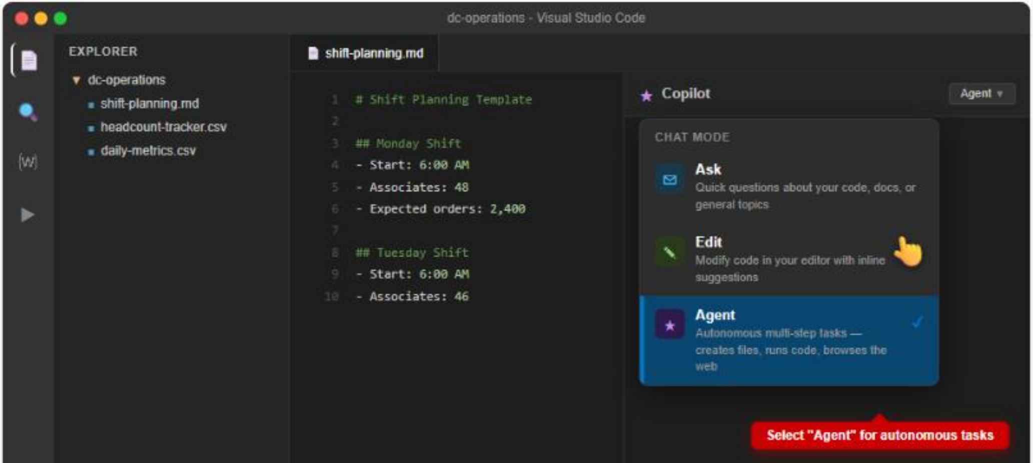
Iterate: **5 min**

Discussion: **3 min**

EXERCISE 4 — SETUP

Setup: Switch to Agent Mode

ⓘ **POLICY REMINDER:** Use generic data only — no Tesla metrics, employee names, or internal processes



dc-operations - Visual Studio Code

EXPLORER

- dc-operations
  - shift-planning.md
  - headcount-tracker.csv
  - daily-metrics.csv

shift-planning.md

```
# Shift Planning Template

## Monday Shift
- Start: 6:00 AM
- Associates: 48
- Expected orders: 2,400

## Tuesday Shift
- Start: 6:00 AM
- Associates: 46
```

Copilot — Agent ▾

CHAT MODE

Ask
Quick questions about your code, docs, or general topics

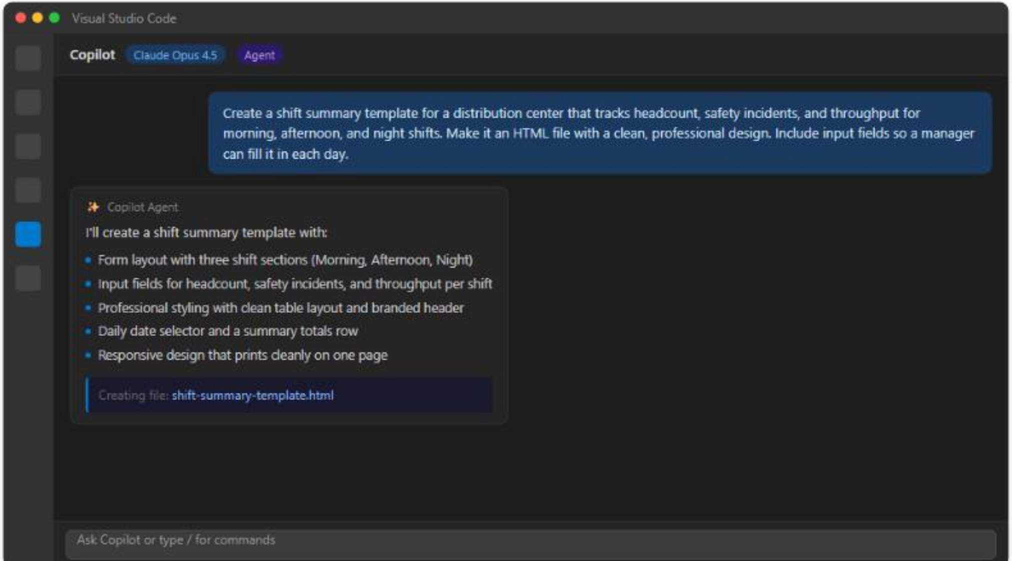
Edit
Modify code in your editor with inline suggestions

Agent ✓
Autonomous multi-step tasks — creates files, runs code, browses the web

Select "Agent" for autonomous tasks

EXERCISE 4 — YOUR FIRST AGENT

TYPE THIS: Your First Agent Task



Create a shift summary template for a distribution center that tracks headcount, safety incidents, and throughput for morning, afternoon, and night shifts. Make it an HTML file with a clean, professional design. Include input fields so a manager can fill it in each day.

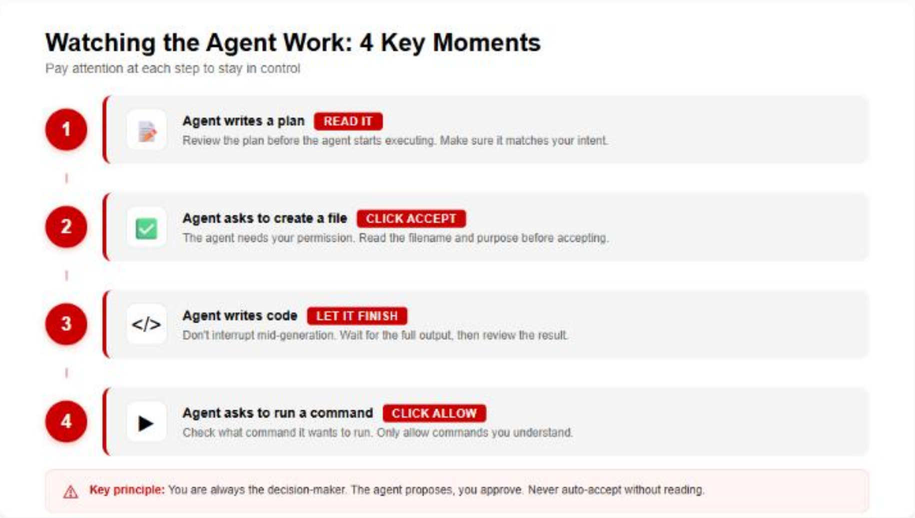
I'll create a shift summary template with:

- Form layout with three shift sections (Morning, Afternoon, Night)
- Input fields for headcount, safety incidents, and throughput per shift
- Professional styling with clean table layout and branded header
- Daily date selector and a summary totals row
- Responsive design that prints cleanly on one page

Creating file: shift-summary-template.html

Ask Copilot or type / for commands

What to Watch For



Watching the Agent Work: 4 Key Moments

Pay attention at each step to stay in control

1. **Agent writes a plan** **READ IT**
   Review the plan before the agent starts executing. Make sure it matches your intent.

2. **Agent asks to create a file** **CLICK ACCEPT**
   The agent needs your permission. Read the filename and purpose before accepting.

3. **Agent writes code** **LET IT FINISH**
   Don't interrupt mid-generation. Wait for the full output, then review the result.

4. **Agent asks to run a command** **CLICK ALLOW**
   Check what command it wants to run. Only allow commands you understand.

⚠ **Key principle:** You are always the decision-maker. The agent proposes, you approve. Never auto-accept without reading.

EXERCISE 4 — REVIEW

Review and Iterate

DO THIS

1. Find the file in VS Code **file explorer** (left sidebar)
2. Right-click the HTML file > **"Open in Default Browser"**
3. Check: Fields for all three shifts? Headcount, safety, throughput? **Clean look?**

THEN PICK ONE IMPROVEMENT:

A

"Add a section that automatically calculates total headcount and throughput across all shifts"

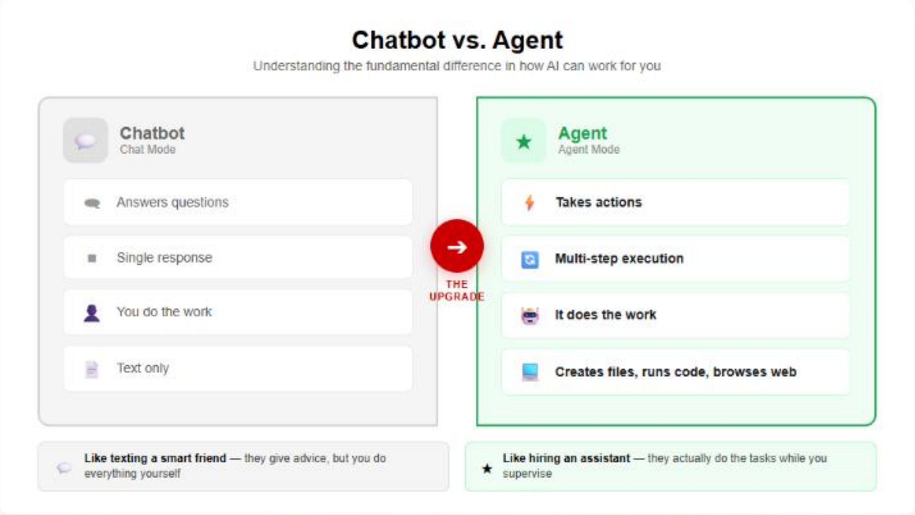
B

"Add a notes field for each shift for supervisor comments"

C

"Change the color scheme to dark background with red accents"

What Just Happened



Chatbot vs. Agent

Understanding the fundamental difference in how AI can work for you

Chatbot Chat Mode	**Agent** Agent Mode
Answers questions	**Takes actions**
Single response	**Multi-step execution**
You do the work	**It does the work**
Text only	**Creates files, runs code, browses web**
Like texting a smart friend — they give advice, but you do everything yourself	★ **Like hiring an assistant** — they actually do the tasks while you supervise

You just went from chatting with AI to commanding AI.

BREAK

10 minutes

Back at 2:10

09

SECTION 9

MCP Tools: Giving Agents Skills

5 minutes

Concept Intro

Extending agent capabilities

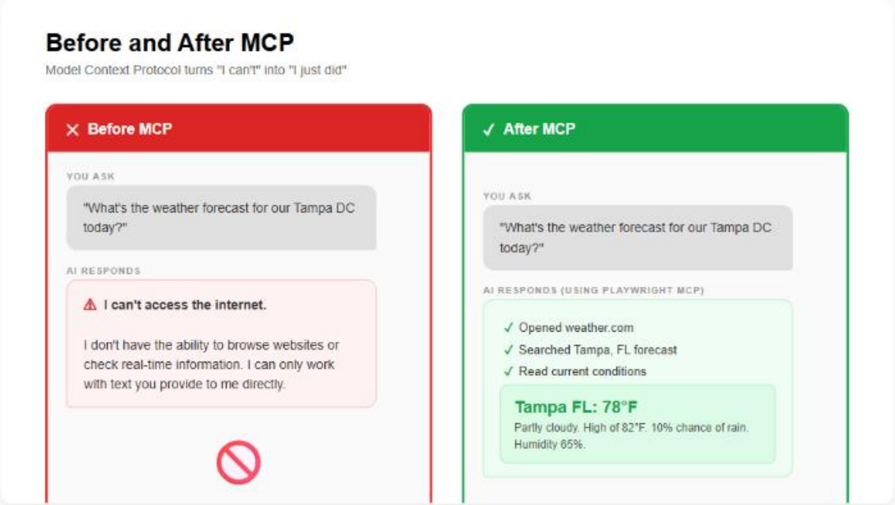
The Big Idea: Agents Can Learn New Skills

- Right now, your Copilot agent can: **read files, write files, run commands.** That's it.
- **MCP** (Model Context Protocol) gives agents new abilities
- **Think of it as installing apps on your phone**
- Each MCP tool = a new skill the agent can use
- Created by Anthropic in November 2024, now an open industry standard
- Over **10,000 MCP tools** available and growing fast

MCP = Model Context Protocol

Open standard · Anthropic 2024

The Phone Analogy



Before and After MCP

Model Context Protocol turns "I can't" into "I just did"

✕ Before MCP

YOU ASK

"What's the weather forecast for our Tampa DC today?"

AI RESPONDS

⚠ **I can't access the internet.**

I don't have the ability to browse websites or check real-time information. I can only work with text you provide to me directly.

✓ After MCP

YOU ASK

"What's the weather forecast for our Tampa DC today?"

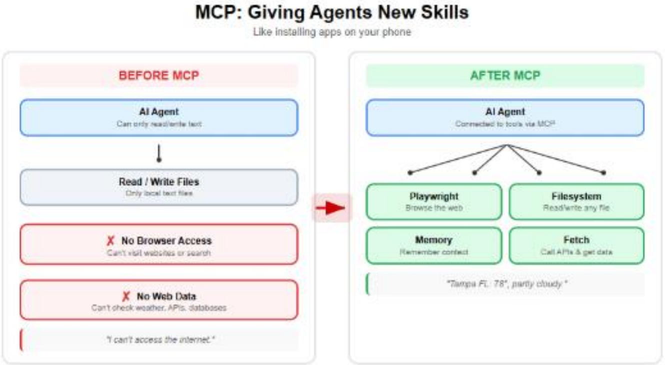
AI RESPONDS (USING PLAYWRIGHT MCP)

- ✓ Opened weather.com
- ✓ Searched Tampa, FL forecast
- ✓ Read current conditions

Tampa FL: 78°F

Partly cloudy. High of 82°F. 10% chance of rain. Humidity 65%.

MCP: Giving Agents New Skills



MCP: Giving Agents New Skills

Like installing apps on your phone

BEFORE MCP

AI Agent
Can only read/write text

Read / Write Files
Only local text files

✗ **No Browser Access**
Can't visit websites or search

✗ **No Web Data**
Can't check weather, APIs, databases

"I can't access the internet."

AFTER MCP

AI Agent
Connected to tools via MCP

Playwright
Browse the web

Filesystem
Read/write any file

Memory
Remember context

Fetch
Call APIs & get data

"Tampa FL: 78°, partly cloudy."

BEFORE MCP

Agent can only read/write files. No browser. No web data.

AFTER MCP

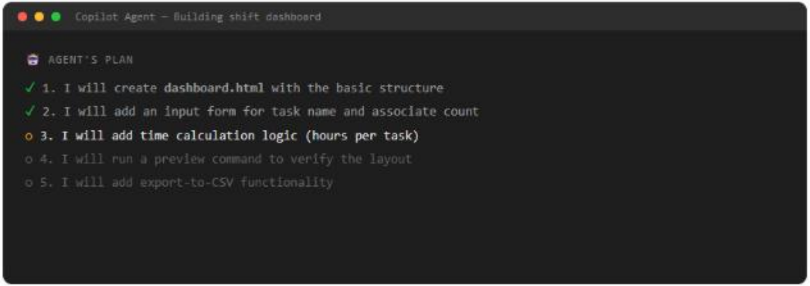
Agent connects to Playwright (browse), Filesystem (organize), Memory (remember), Fetch (connect)

Each MCP tool = a new skill. Before: limited to files. After: connected to the world.

Before MCP vs After MCP

Reading an Agent's Plan

Always review what the agent intends to do before it does it



```
Copilot Agent — Building shift dashboard

🤖 AGENT'S PLAN
✓ 1. I will create dashboard.html with the basic structure
✓ 2. I will add an input form for task name and associate count
○ 3. I will add time calculation logic (hours per task)
○ 4. I will run a preview command to verify the layout
○ 5. I will add export-to-CSV functionality
```

YOUR RESPONSE OPTIONS

✓ "Looks good, continue" ✎ "Change step 3 to use minutes instead" ✕ "Stop — wrong approach"

Tip: Intervene early. It's easier to redirect an agent before it builds the wrong thing.

Types of MCP Skills

Playwright

Browse any website

Pull data from public dashboards

Filesystem

Organize files and folders

Create folder structures automatically

Memory

Remember facts between sessions

Store your DC's target KPIs

Fetch

Connect to web services

Pull data from APIs

28% of Fortune 500 companies already use MCP

SECTION 10

Exercise 5: Install & Use First MCP Skill

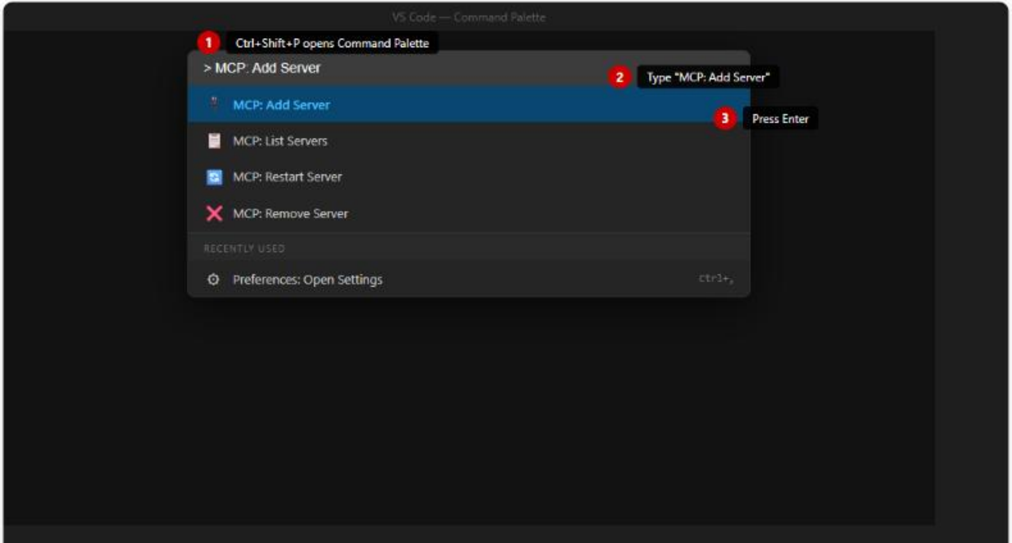
25 minutes

Install: 5 min | First task: 10 min | Compare: 5 min | Discuss: 5 min

⚠ **Policy Reminder:** Visit only public websites. Do NOT navigate to Tesla internal sites.

EXERCISE 5 -- STEP 1

Step 1: Install Playwright MCP



1. Ctrl+Shift+P opens Command Palette
2. Type "MCP: Add Server"
3. Press Enter

Troubleshooting MCP Install

⚠ **"npx is not recognized"**

➔ Install Node.js from nodejs.org, restart VS Code, try again

⚠ **Server fails to start**

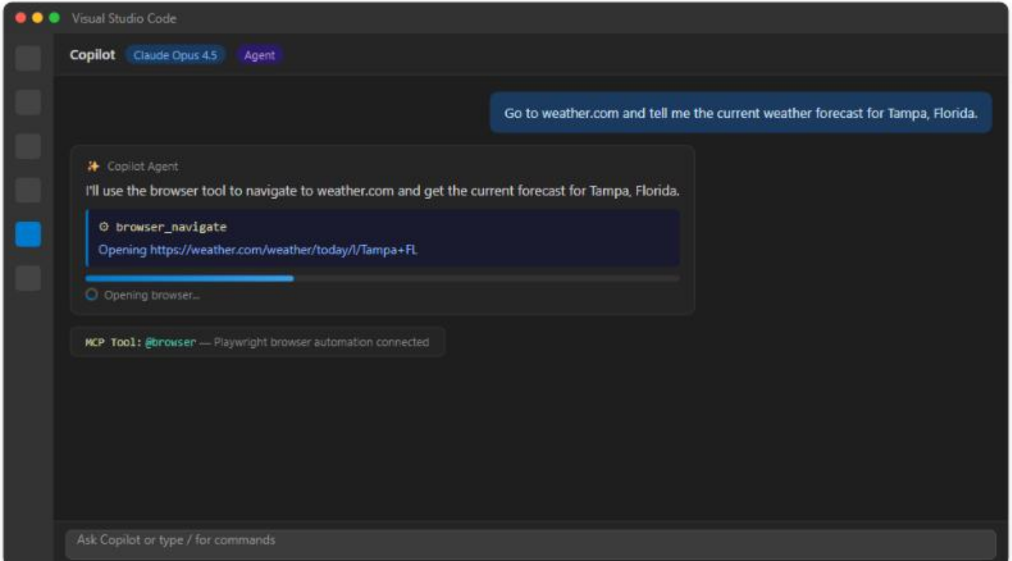
➔ Check internet connection, try the command again

⚠ **No MCP option in Command Palette**

➔ Update VS Code to latest version (Help > Check for Updates). VS Code 1.99+ required.

EXERCISE 5 — STEP 2

Step 2: Use the New Skill



Go to weather.com and tell me the current weather forecast for Tampa, Florida.

I'll use the browser tool to navigate to weather.com and get the current forecast for Tampa, Florida.

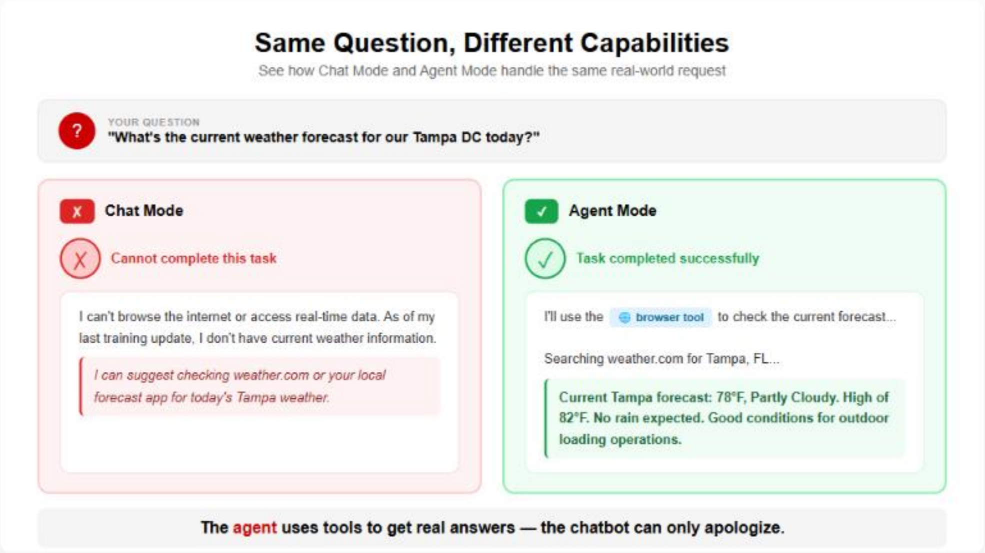
```
⚙ browser_navigate
Opening https://weather.com/weather/today/l/Tampa+FL
```

○ Opening browser...

`MCP Tool: @browser` — Playwright browser automation connected

EXERCISE 5 — STEP 3

Step 3: Compare Before vs After



Same Question, Different Capabilities

See how Chat Mode and Agent Mode handle the same real-world request

?

YOUR QUESTION

"What's the current weather forecast for our Tampa DC today?"

✗ **Chat Mode**	✓ **Agent Mode**
✗ Cannot complete this task	✓ Task completed successfully
I can't browse the internet or access real-time data. As of my last training update, I don't have current weather information. *I can suggest checking weather.com or your local forecast app for today's Tampa weather.*	I'll use the 🌐 browser tool to check the current forecast... Searching weather.com for Tampa, FL... **Current Tampa forecast: 78°F, Partly Cloudy. High of 82°F. No rain expected. Good conditions for outdoor loading operations.**

The agent uses tools to get real answers — the chatbot can only apologize.

What Other MCP Skills Would Help?

Brainstorm: what tools would save time at your distribution center?

- A tool that reads spreadsheets and CSVs?
- A tool that connects to a database?
- A tool that sends formatted emails?
- A tool that monitors a website for changes?

Think about what would save the most time at your DC

SECTION 11

Reading Agent Plans & Git Basics

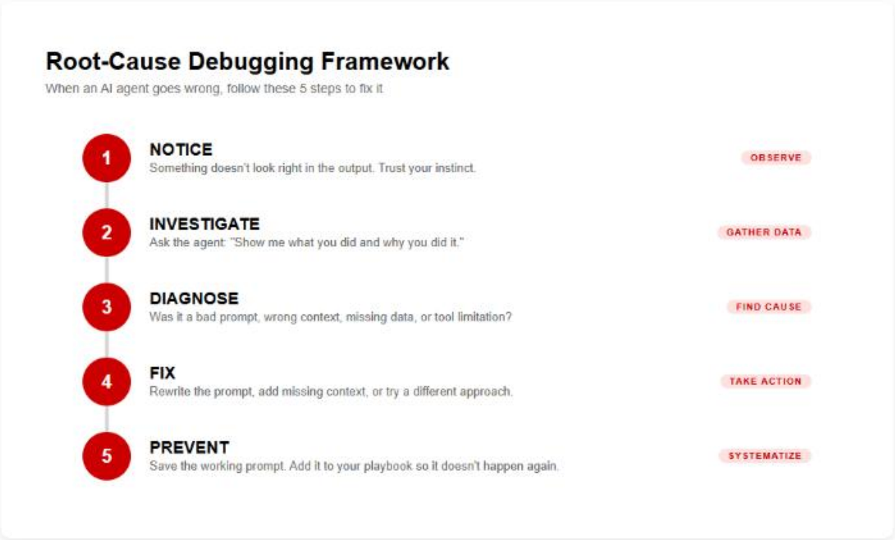
5 minutes

Concept Intro

Understanding agent plans and version control

Reading an Agent’s Plan

Before an agent acts, it tells you what it plans to do. Read it like a to-do list:



Root-Cause Debugging Framework

When an AI agent goes wrong, follow these 5 steps to fix it

1 **NOTICE**
Something doesn’t look right in the output. Trust your instinct.
OBSERVE

2 **INVESTIGATE**
Ask the agent: "Show me what you did and why you did it."
GATHER DATA

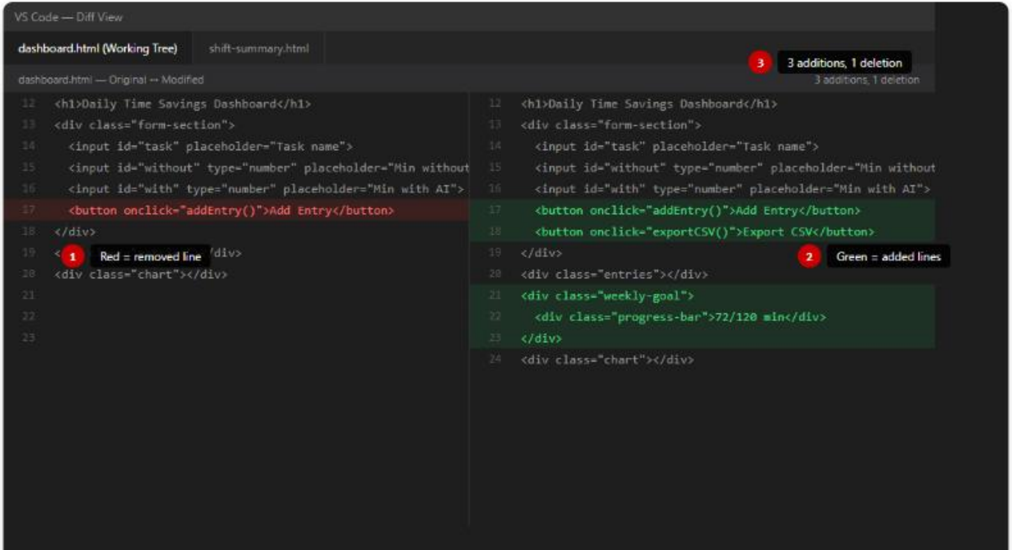
3 **DIAGNOSE**
Was it a bad prompt, wrong context, missing data, or tool limitation?
FIND CAUSE

4 **FIX**
Rewrite the prompt, add missing context, or try a different approach.
TAKE ACTION

5 **PREVENT**
Save the working prompt. Add it to your playbook so it doesn’t happen again.
SYSTEMATIZE

What Is a Diff?

"Track Changes" for Code


VS Code — Diff View
dashboard.html (Working Tree)
shift-summary.html
dashboard.html — Original ↔ Modified
3 additions, 1 deletion

```
<h1>Daily Time Savings Dashboard</h1>
<div class="form-section">
  <input id="task" placeholder="Task name">
  <input id="without" type="number" placeholder="Min without
  <input id="with" type="number" placeholder="Min with AI">
  <button onclick="addEntry()">Add Entry</button>
</div>
<                  /div>
<div class="chart"></div>
```

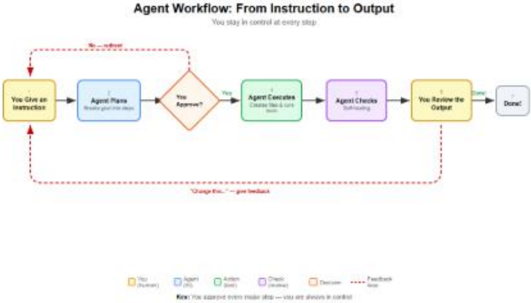
```
<h1>Daily Time Savings Dashboard</h1>
<div class="form-section">
  <input id="task" placeholder="Task name">
  <input id="without" type="number" placeholder="Min without
  <input id="with" type="number" placeholder="Min with AI">
  <button onclick="addEntry()">Add Entry</button>
  <button onclick="exportCSV()">Export CSV</button>
</div>
<div class="entries"></div>
<div class="weekly-goal">
  <div class="progress-bar">72/120 min</div>
</div>
<div class="chart"></div>
```

① Red = removed line

② Green = added lines

③ 3 additions, 1 deletion

The Review Workflow



1. Agent proposes changes (shows you the plan)
2. You review the diff (green = added, red = removed)
3. You accept or reject
4. If you reject, tell the agent what to do differently

KEY RULE

"Never accept changes you do not understand. Ask: 'Explain what you changed and why.'"

How to Undo

METHOD 1

Keyboard Shortcut

`Ctrl + Z`

Undo the last change. Works in any file, any time.

METHOD 2

Tell the Agent

`"Undo those changes"`

Or say: "Revert that file." The agent will reverse its own edits.

METHOD 3

Source Control Panel

`Click ↺ next to file`

VS Code Source Control panel: click the undo icon next to any changed file.

You can always undo. Nothing is permanent.

SECTION 12

Exercise 6: Agent Builds a Real Tool

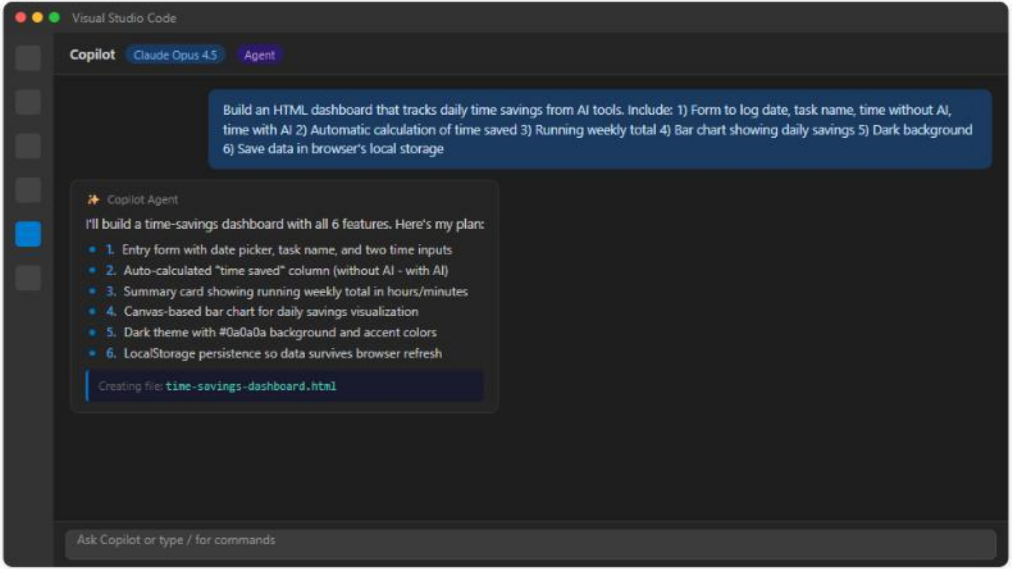
30 minutes

Describe: **3 min**
Agent builds: **12 min**
Test: **5 min**
Iterate: **5 min**
Review diff: **5 min**

⚠ **Policy Reminder:** Use generic/fictional data only

EXERCISE 6 — BUILD A DASHBOARD

TYPE THIS: Build a Dashboard



Build an HTML dashboard that tracks daily time savings from AI tools. Include: 1) Form to log date, task name, time without AI, time with AI 2) Automatic calculation of time saved 3) Running weekly total 4) Bar chart showing daily savings 5) Dark background 6) Save data in browser's local storage

I'll build a time-savings dashboard with all 6 features. Here's my plan:

1. Entry form with date picker, task name, and two time inputs
2. Auto-calculated "time saved" column (without AI - with AI)
3. Summary card showing running weekly total in hours/minutes
4. Canvas-based bar chart for daily savings visualization
5. Dark theme with #0a0a0a background and accent colors
6. LocalStorage persistence so data survives browser refresh

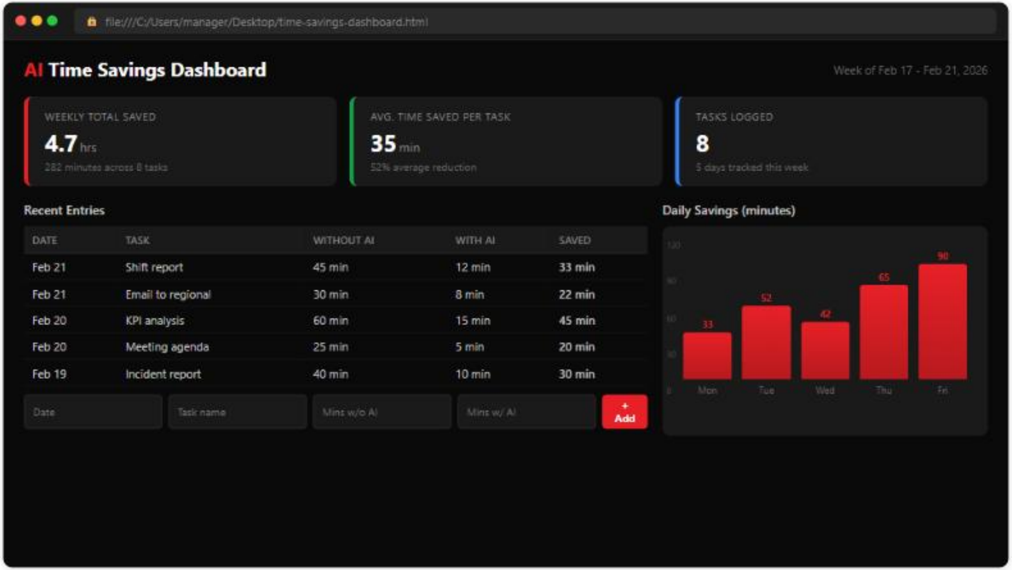
Creating file: `time-savings-dashboard.html`

What to Watch the Agent Do

1. Agent creates a **plan** — does it cover all 6 requirements?
2. Agent asks to create a file (like **time-savings-dashboard.html**) — click **“Accept”**
3. Agent writes **HTML, CSS, and JavaScript** — let it finish (2–3 min)
4. Agent may **iterate on its own** — if it spots an error, it fixes it automatically
5. This is **“self-healing” behavior** — the agent **checks its own work**

EXERCISE 6 — TEST

Test It in Your Browser


file:///C:/Users/manager/Desktop/time-savings-dashboard.html

AI Time Savings Dashboard

Week of Feb 17 - Feb 21, 2026

WEEKLY TOTAL SAVED
4.7 hrs
282 minutes across 8 tasks

AVG. TIME SAVED PER TASK
35 min
52% average reduction

TASKS LOGGED
8
5 days tracked this week

Recent Entries

DATE	TASK	WITHOUT AI	WITH AI	SAVED
Feb 21	Shift report	45 min	12 min	33 min
Feb 21	Email to regional	30 min	8 min	22 min
Feb 20	KPI analysis	60 min	15 min	45 min
Feb 20	Meeting agenda	25 min	5 min	20 min
Feb 19	Incident report	40 min	10 min	30 min

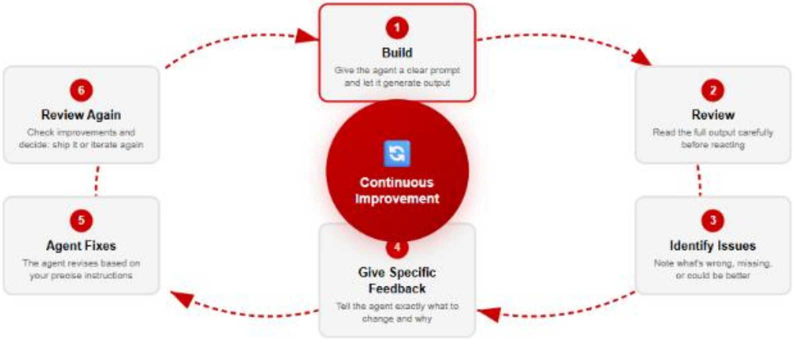
Daily Savings (minutes)

EXERCISE 6 — ITERATE

Ask for an Improvement

The Iteration Cycle

Great results come from refining, not from the first attempt



1
Build
Give the agent a clear prompt and let it generate output

2
Review
Read the full output carefully before reacting

3
Identify Issues
Note what's wrong, missing, or could be better

4
Give Specific Feedback
Tell the agent exactly what to change and why

5
Agent Fixes
The agent revises based on your precise instructions

6
Review Again
Check improvements and decide: ship it or iterate again

Continuous Improvement

Pro tip: Most AGMs get great results by the 2nd or 3rd iteration. Each round of specific feedback gets you 80% closer to perfect.

Review What Changed (The Diff)

1. Look at the file in VS Code — colored markers in the left margin (green = added, red = removed)
2. Click the **Source Control** icon in the left sidebar (branch/fork icon)
3. Click the **changed file** to see the full diff

TWO QUESTIONS TO ASK YOURSELF

- “Can you identify the new code for your requested feature?”
- “Did the agent change anything you did NOT ask for?”

The agent did great work. But what happens when it doesn't?

13

SECTION 13

When Agents Don’t Deliver

5 minutes | Concept Intro

Real-world case study — root-cause analysis

AGENT QUALITY CONTROL

The Root Cause Framework

You Already Know How to Do This

Managing AI agents uses the same skills you use managing your team

#	👥 Managing People		🤖 Managing Agents
1	**Check the work** at each station	→	**Check the output** at each step
2	**Break projects** into milestones	→	**Break tasks** into batches
3	**Quality gate** before shipping	→	**Validate** before moving on
4	**Give clear instructions**, not vague goals	→	**Write specific prompts**, not vague asks
5	**Redirect** when off track	→	**Course-correct** when output drifts
6	**Standard procedures** for consistency	→	**Saved prompts** for repeatability

Key insight: Your leadership experience is your biggest advantage. Managing agents is managing work -- you're already an expert.

CASE STUDY

Real Case Study: The 75-Slide Problem

TASK
"Generate 75 slides with professional graphics and branded visuals"

✓ AGENT REPORTED
"Done! 75 slides created and committed."

✗ REALITY
All 75 were plain text on black backgrounds. Zero graphics. Zero visuals.

⚠ ROOT CAUSE
Task too big. Agent spent all capacity writing text, never got to quality steps.

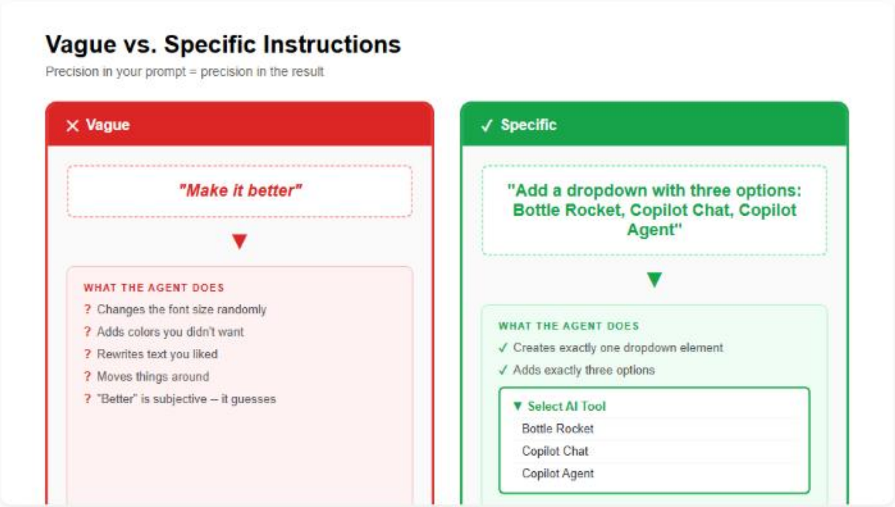
FIX 1: RIGHT-SIZE
Break into batches of 5 slides

FIX 2: QUALITY-FIRST
Validate each batch BEFORE moving to the next

Lesson: **"Quality inspection at each station, not end of line."**

KEY INSIGHT

Managing Agents = Managing People



Vague vs. Specific Instructions

Precision in your prompt = precision in the result

✕ Vague

"Make it better"

▼

WHAT THE AGENT DOES

- ? Changes the font size randomly
- ? Adds colors you didn't want
- ? Rewrites text you liked
- ? Moves things around
- ? "Better" is subjective – it guesses

✓ Specific

"Add a dropdown with three options: Bottle Rocket, Copilot Chat, Copilot Agent"

▼

WHAT THE AGENT DOES

- ✓ Creates exactly one dropdown element
- ✓ Adds exactly three options

▼ Select AI Tool
Bottle Rocket
Copilot Chat
Copilot Agent

13b

SECTION

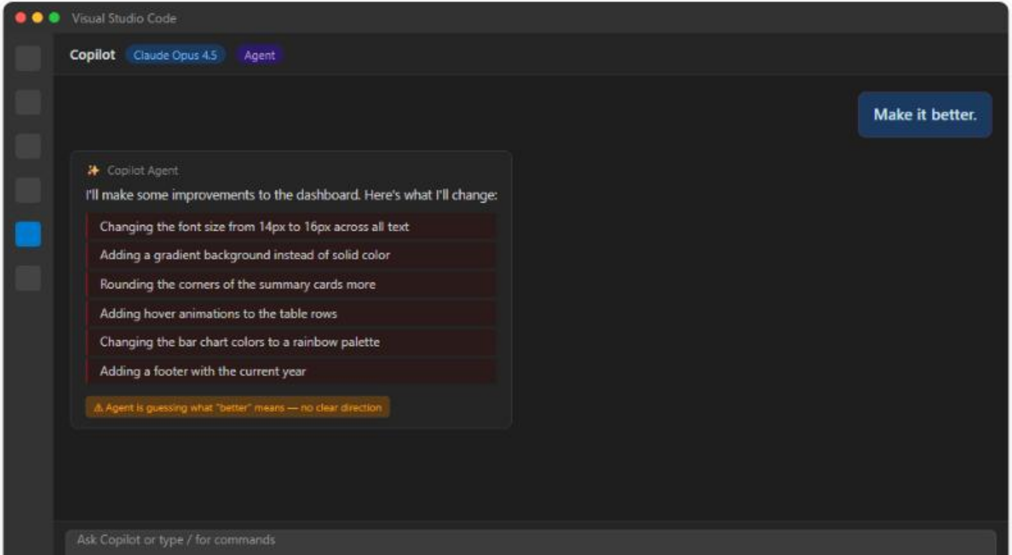
Exercise 7: Debug & Redirect an Agent

PART A: VAGUE INSTRUCTIONS	PART B: CATCH A MISTAKE
10 min	**15 min**

Total: **25 minutes**

EXERCISE 7 -- PART A

Part A: What Happens with Vague Instructions



Make it better.

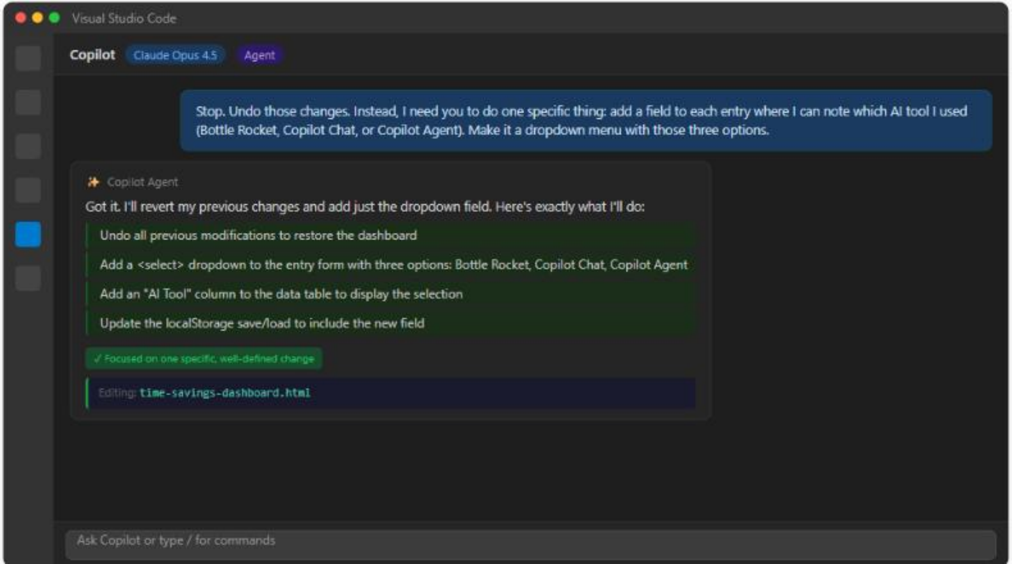
I'll make some improvements to the dashboard. Here's what I'll change:

- Changing the font size from 14px to 16px across all text
- Adding a gradient background instead of solid color
- Rounding the corners of the summary cards more
- Adding hover animations to the table rows
- Changing the bar chart colors to a rainbow palette
- Adding a footer with the current year

⚠ Agent is guessing what "better" means — no clear direction

EXERCISE 7 — REDIRECT

Now Redirect with Specific Instructions



Stop. Undo those changes. Instead, I need you to do one specific thing: add a field to each entry where I can note which AI tool I used (Bottle Rocket, Copilot Chat, or Copilot Agent). Make it a dropdown menu with those three options.

Copilot Agent

Got it. I'll revert my previous changes and add just the dropdown field. Here's exactly what I'll do:

- Undo all previous modifications to restore the dashboard
- Add a <select> dropdown to the entry form with three options: Bottle Rocket, Copilot Chat, Copilot Agent
- Add an "AI Tool" column to the data table to display the selection
- Update the localStorage save/load to include the new field

✓ Focused on one specific, well-defined change

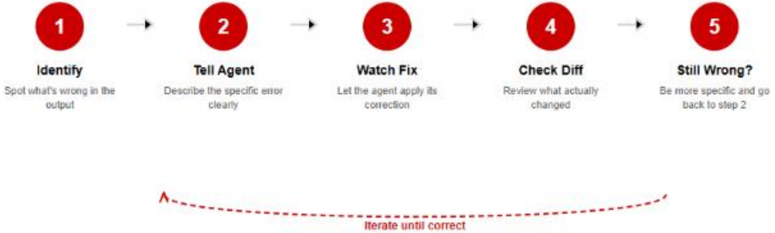
Editing: `time-savings-dashboard.html`

KEY PRINCIPLE

The Lesson: Specific Beats Vague

The Fix Workflow

5 steps to correct any AI agent mistake



1. **Identify**
Spot what's wrong in the output

2. **Tell Agent**
Describe the specific error clearly

3. **Watch Fix**
Let the agent apply its correction

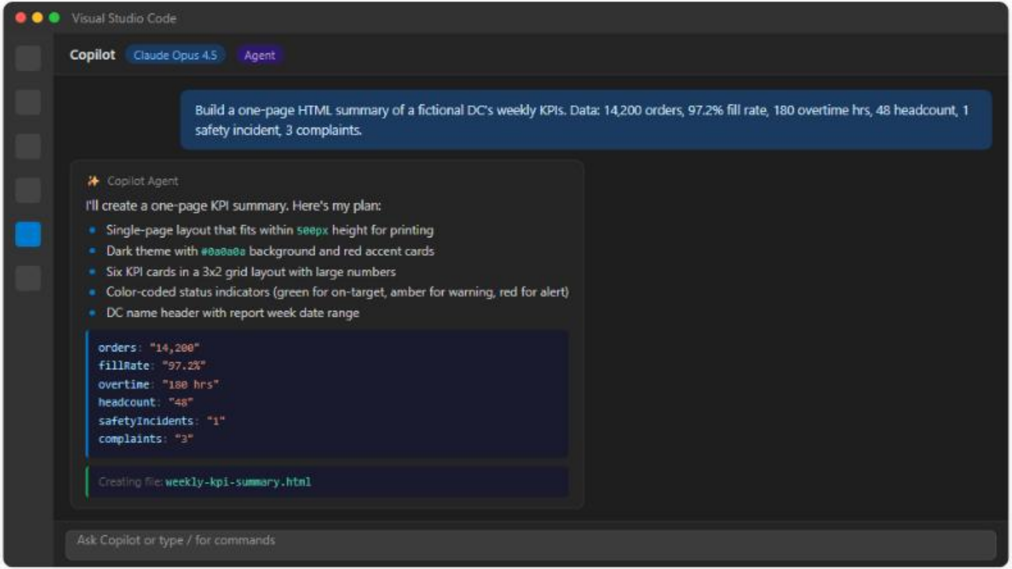
4. **Check Diff**
Review what actually changed

5. **Still Wrong?**
Be more specific and go back to step 2

Remember: Each iteration should include MORE specific language. "Make it better" wastes cycles. "Change the header color to #cc0000" gets it done.

EXERCISE 7 — PART B

Part B: Catch and Fix a Mistake



Build a one-page HTML summary of a fictional DC's weekly KPIs. Data: 14,200 orders, 97.2% fill rate, 180 overtime hrs, 48 headcount, 1 safety incident, 3 complaints.

I'll create a one-page KPI summary. Here's my plan:

- Single-page layout that fits within `500px` height for printing
- Dark theme with `#0a0a0a` background and red accent cards
- Six KPI cards in a 3x2 grid layout with large numbers
- Color-coded status indicators (green for on-target, amber for warning, red for alert)
- DC name header with report week date range

```
orders: "14,200"
fillRate: "97.2%"
overtime: "180 hrs"
headcount: "48"
safetyIncidents: "1"
complaints: "3"
```

Creating file: `weekly-kpi-summary.html`

AGENT QUALITY CONTROL

The Fix Workflow

1. **Identify**
   What is wrong?
   “The page scrolls — it’s too tall”

▼

2. **Tell the Agent Specifically**
   “Reduce the padding, make cards smaller, remove the footer, fit within 500px”

▼

3. **Watch the Agent Fix It**
   Review the new version carefully

▼

4. **Check the Diff**
   What did the agent actually change?

▼

5. **If Still Wrong, Be Even More Specific**
   “Each card must be exactly 80px tall”

> Do NOT say “try again.” Say exactly what is wrong.

EXERCISE 7 — STRETCH CHALLENGE

Challenge: Week-Over-Week Comparison

If the agent got it right on the first try, add this challenge:

TYPE THIS

> Add a second row of cards showing week-over-week change. Green arrows for improvement, red arrows for decline.

COMPARISON DATA

KPI	LAST WEEK	CHANGE
Orders	13,800	▲ 2.9%
Fill Rate	96.5%	▲ 0.7%
Overtime	210 hrs	▼ 14.3% (good)
Headcount	46	▲ 4.3%
Safety	2 incidents	▼ 50% (good)
Complaints	5	▼ 40% (good)

⚠ Still must fit on one screen without scrolling.

14

SECTION

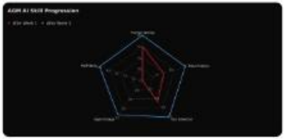
Share Results + Key Takeaways + Homework

SHOW & TELL	TAKEAWAYS	HOMEWORK
12 min	3 min	5 min

Total: **20 minutes**

Five Things to Remember from Today

1. **CoT + Few-Shot + Persona** upgrade your prompts from good to great — always add “step by step” for complex analysis
2. **Agents DO work, chat SUGGESTS work** — that is the fundamental difference
3. **MCP tools give agents new skills** — like installing apps on a phone
4. **Check the output, not the count** — an agent that says “Done!” may have cut corners. Always verify the actual work.
5. **Specific instructions + right-sized tasks** — break big jobs into small batches with checkpoints



Your Skill Progression

Homework

45 minutes over the week — small daily actions, big weekly gains

1 Use Agent Mode for 1 Real Task

NO Tesla data in Copilot. Try a checklist, time tracker, or report layout.

~15 min

2 Track AI Time Savings Daily

Use your dashboard — at least 1 entry/day. Target 20–30 min saved/day.

~5 min/day

3 Explore MCP Tools

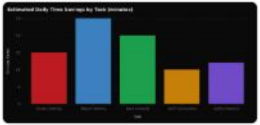
Type "What MCP tools are available?" in Copilot Chat. List 3 that would help your DC work.

~10 min

4 Think Ahead — Capstone Project

What process would you most want to automate? Write 2 sentences.

~10 min



Total: **~45 min** over the week

COMING NEXT

Week 3: Voice Mode & Advanced Agents

- Voice Mode for rapid dictation (3–5x faster than typing)
- Advanced Agent Mode workflows (multi-file projects)
- Multi-skill agent configurations (combining MCP tools)
- Agent file access and workspace context
- Building more complex, multi-page tools

You went from chatting with AI to ***commanding an AI agent*** *— in one afternoon.*