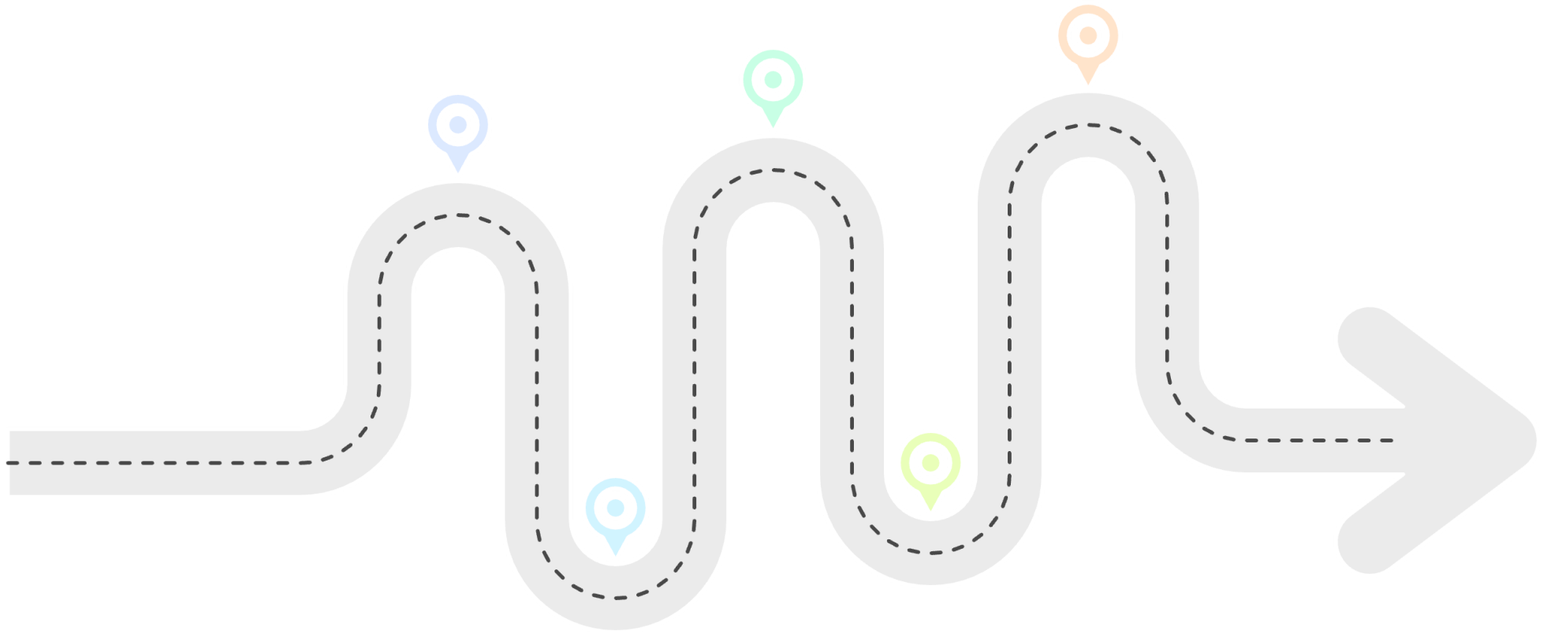
What is GitHub Copilot?

SOFTWARE DEVELOPMENT WITH GITHUB COPILOT

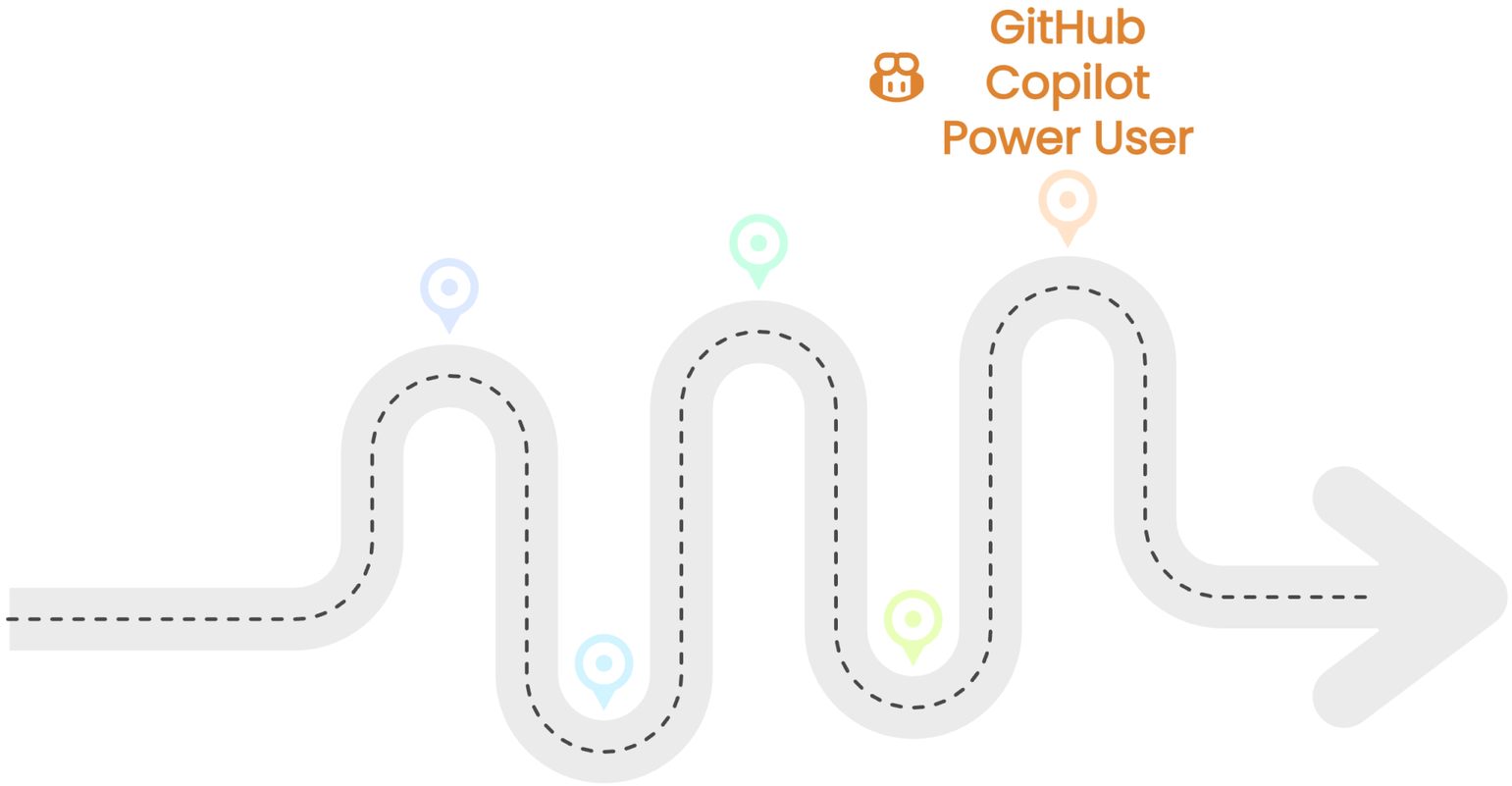
Thalia Barrera
AI Engineering Curriculum Manager, DataCamp

datacamp

The roadmap to power user



The roadmap to power user

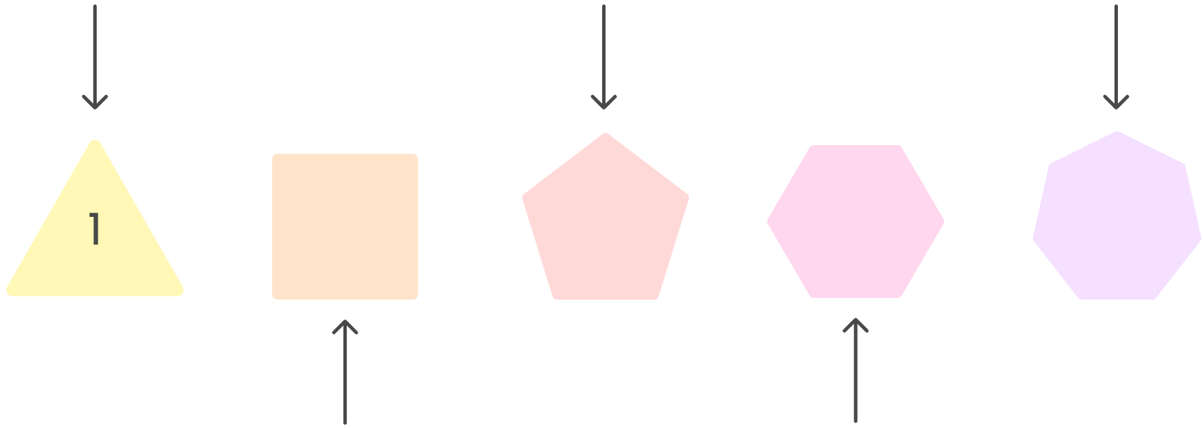


Course learning objectives

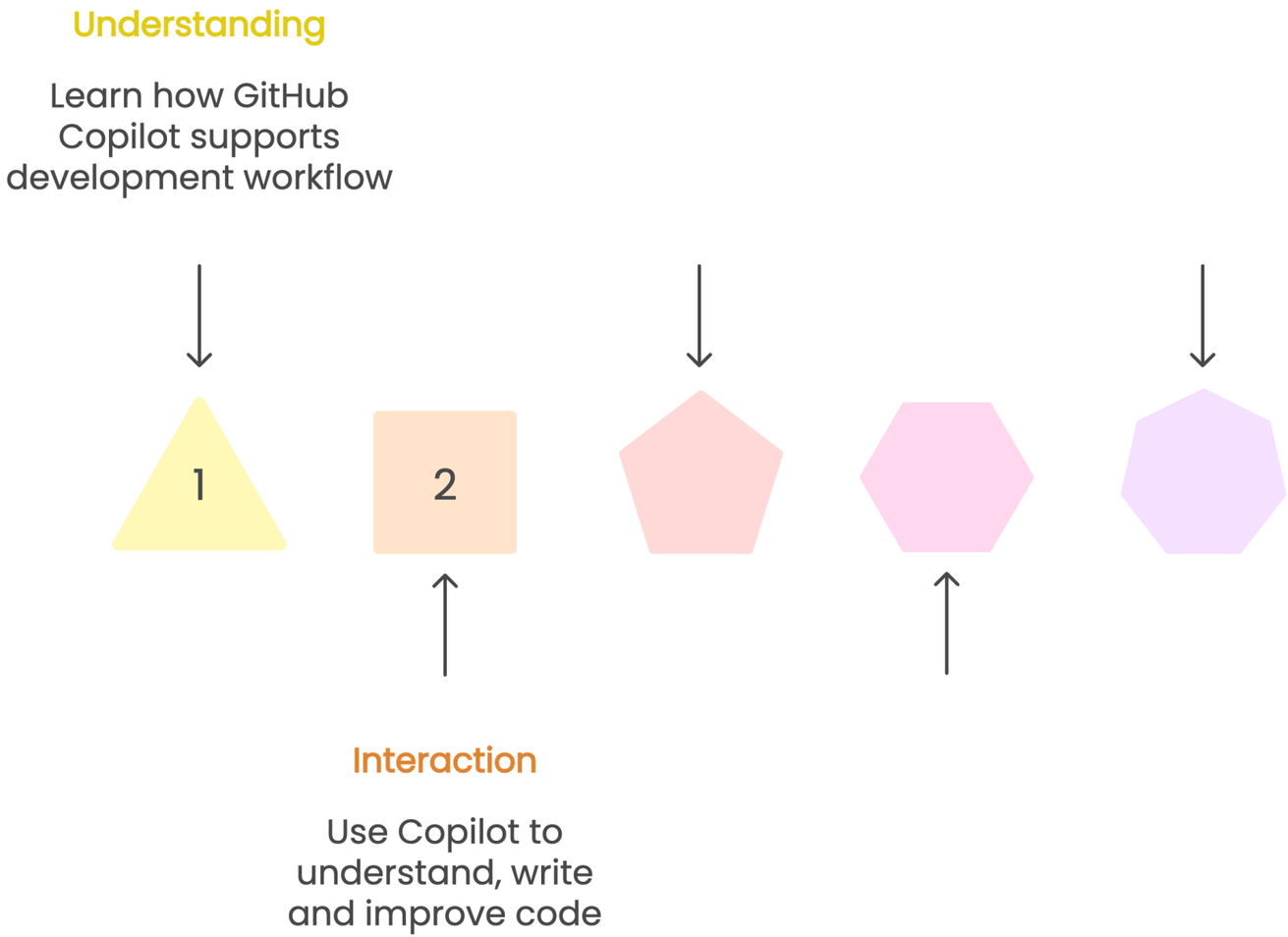
Course learning objectives

Understanding

Learn how GitHub Copilot supports development workflow



Course learning objectives



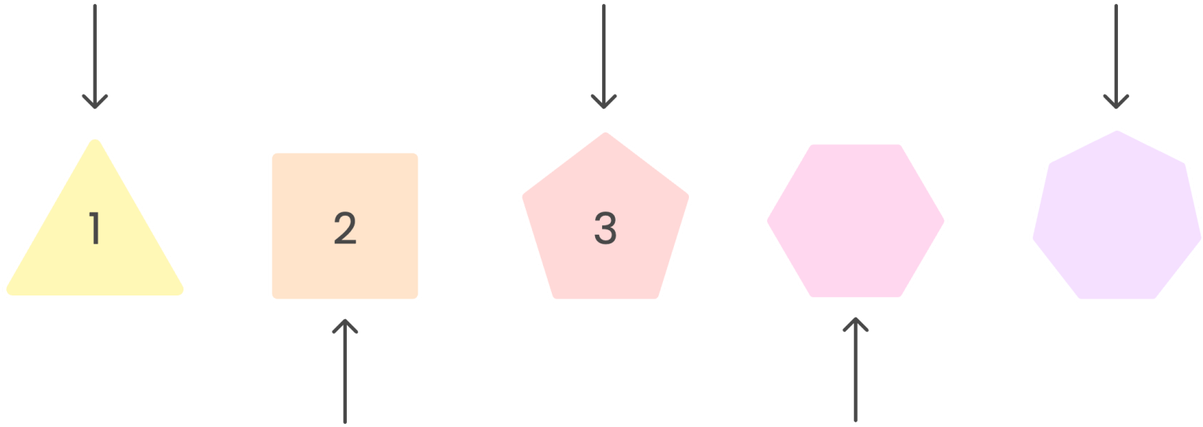
Course learning objectives

Understanding

Learn how GitHub Copilot supports development workflow

Context

Provide accurate context for better responses



Interaction

Use Copilot to understand, write and improve code

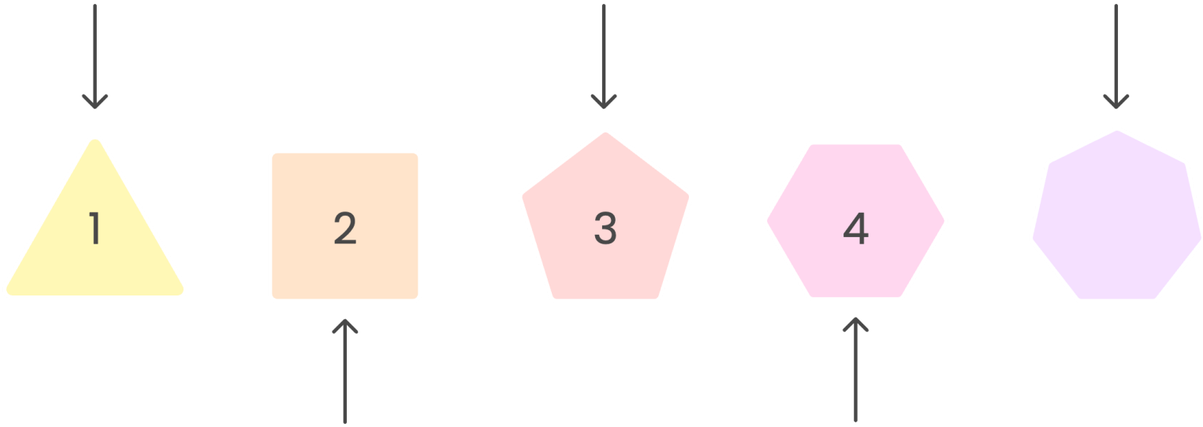
Course learning objectives

Understanding

Learn how GitHub Copilot supports development workflow

Context

Provide accurate context for better responses



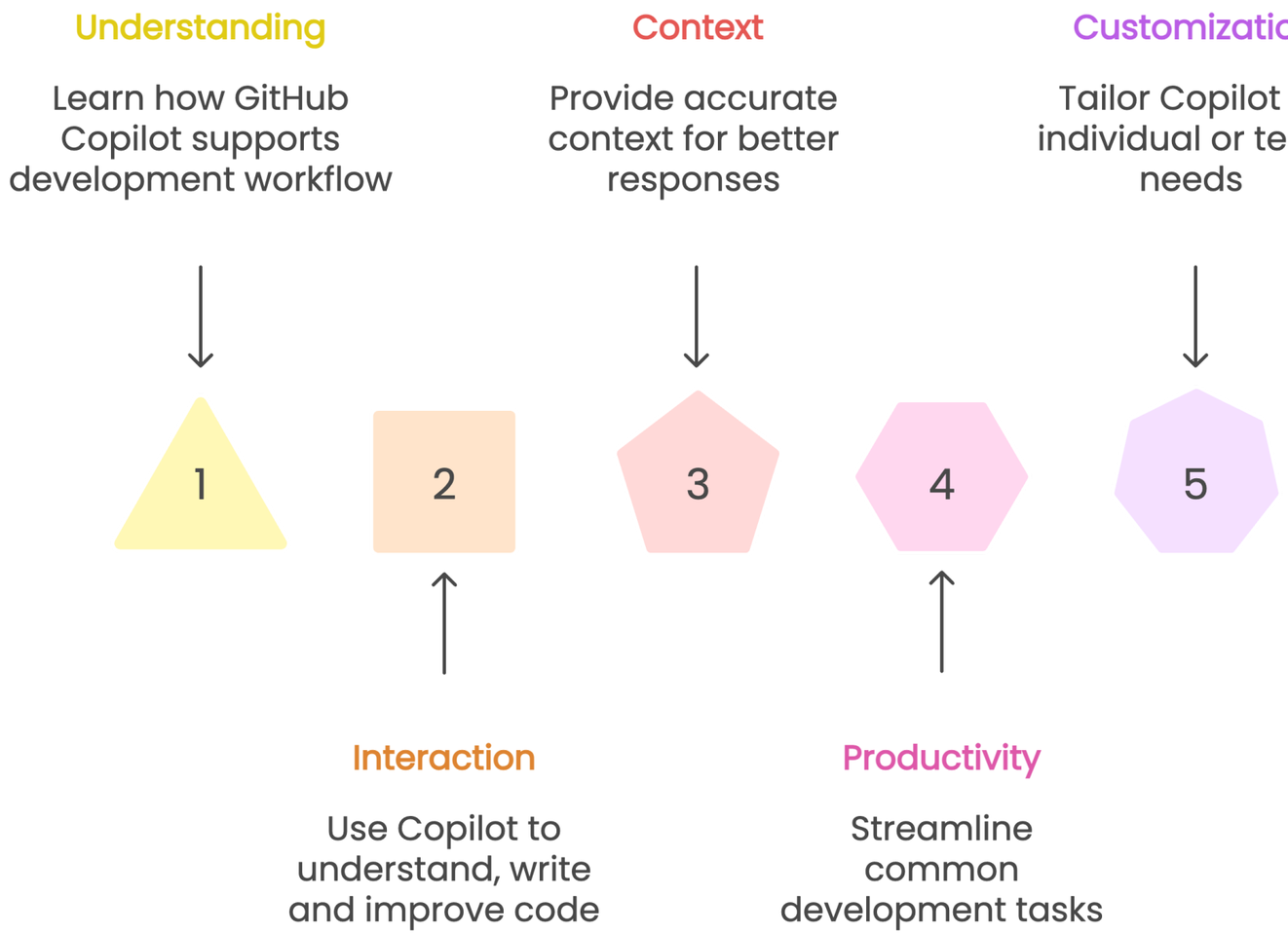
Interaction

Use Copilot to understand, write and improve code

Productivity

Streamline common development tasks

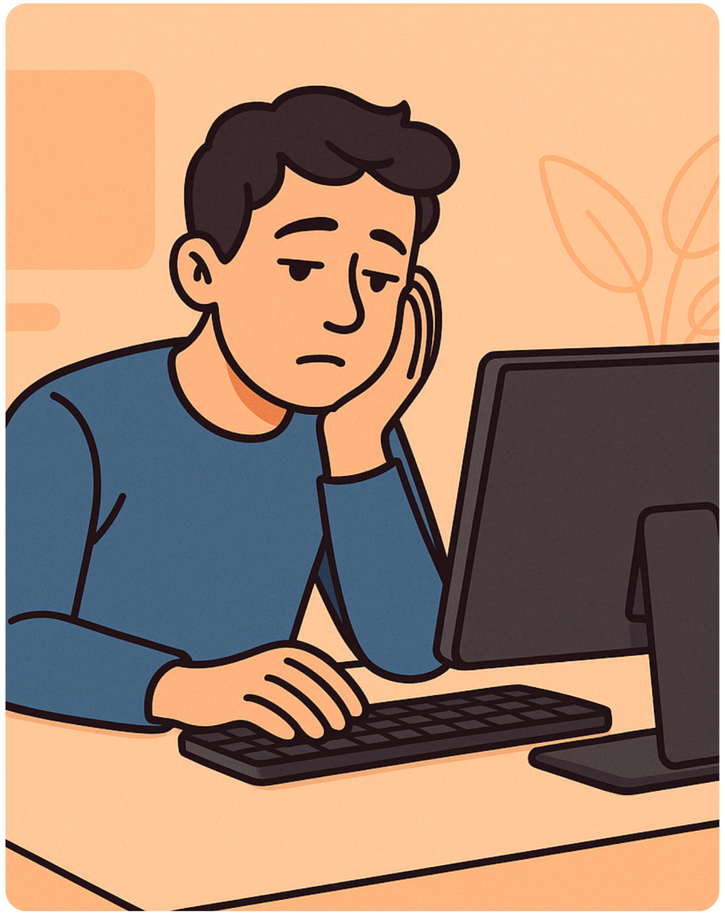
Course learning objectives

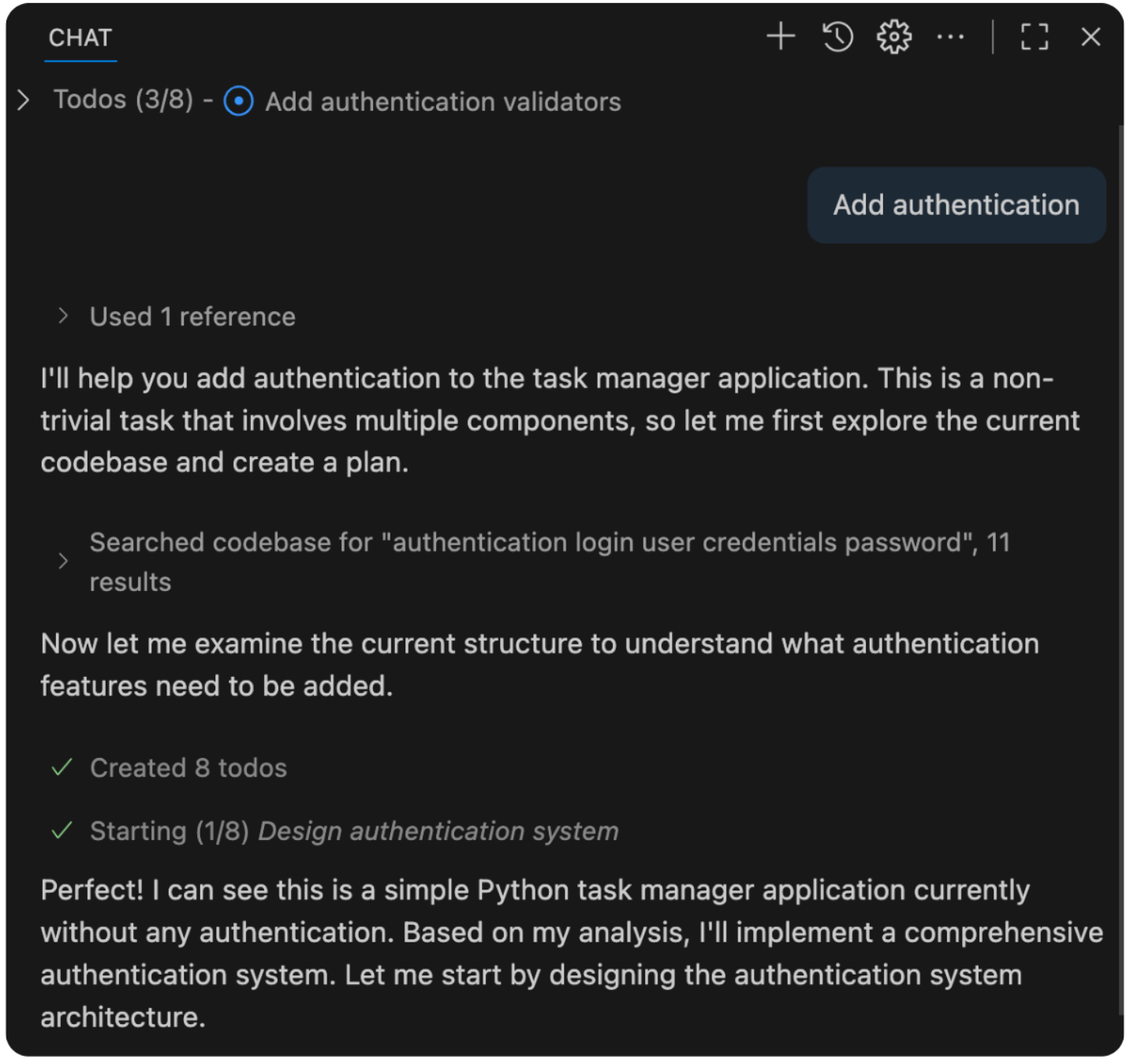


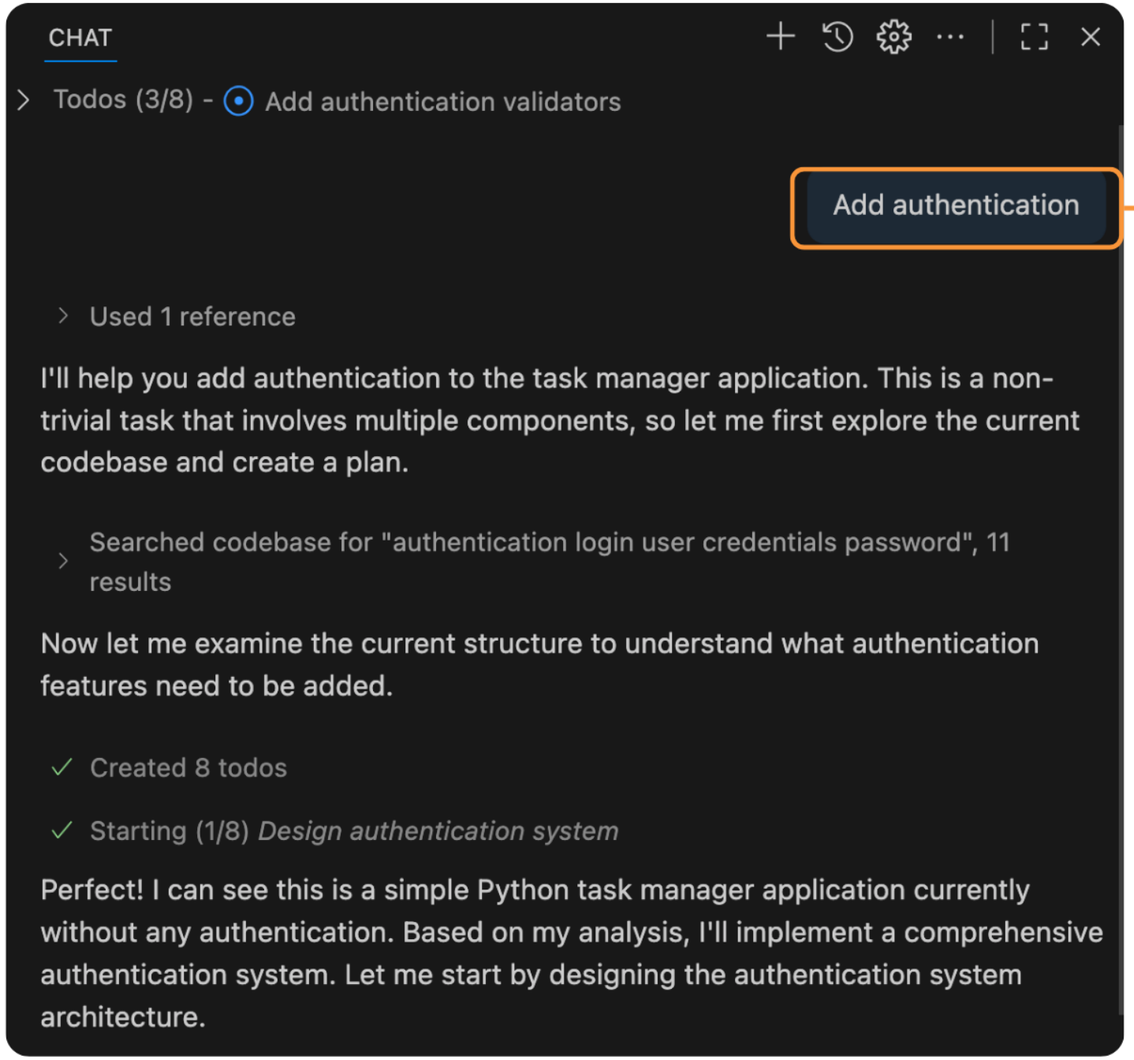
GitHub Copilot

- AI-powered **coding assistant**
- **Understand**, **write**, and **refactor** code using natural language prompts

The GitHub Copilot collaboration spectrum

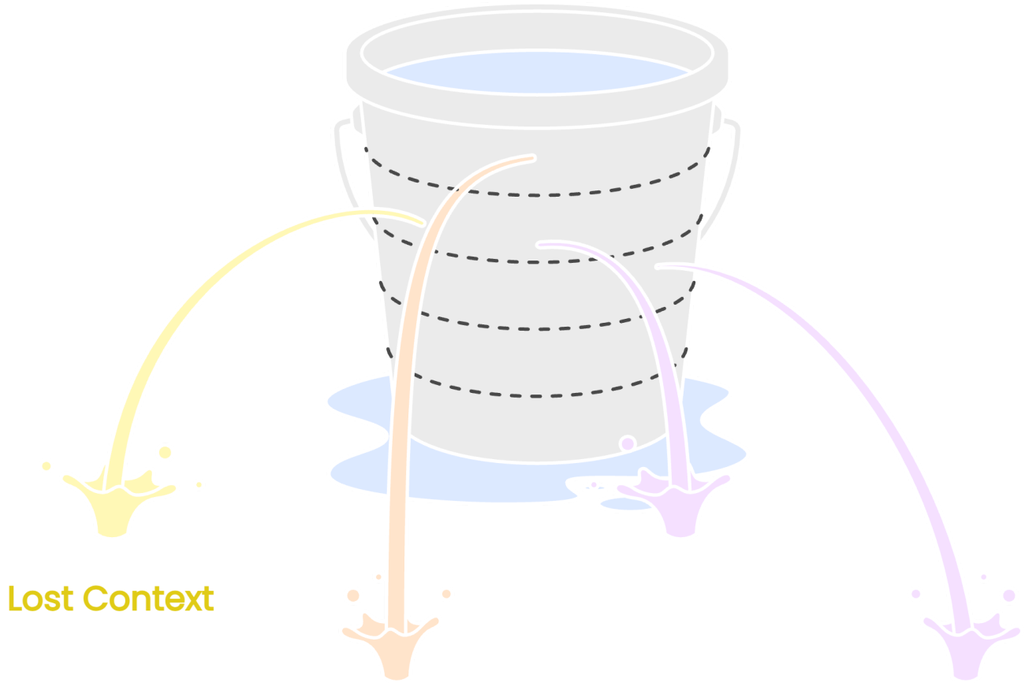



CHAT
Todos (3/8) - Add authentication validators
Add authentication
Used 1 reference
I'll help you add authentication to the task manager application. This is a non-trivial task that involves multiple components, so let me first explore the current codebase and create a plan.
Searched codebase for "authentication login user credentials password", 11 results
Now let me examine the current structure to understand what authentication features need to be added.
Created 8 todos
Starting (1/8) Design authentication system
Perfect! I can see this is a simple Python task manager application currently without any authentication. Based on my analysis, I'll implement a comprehensive authentication system. Let me start by designing the authentication system architecture.

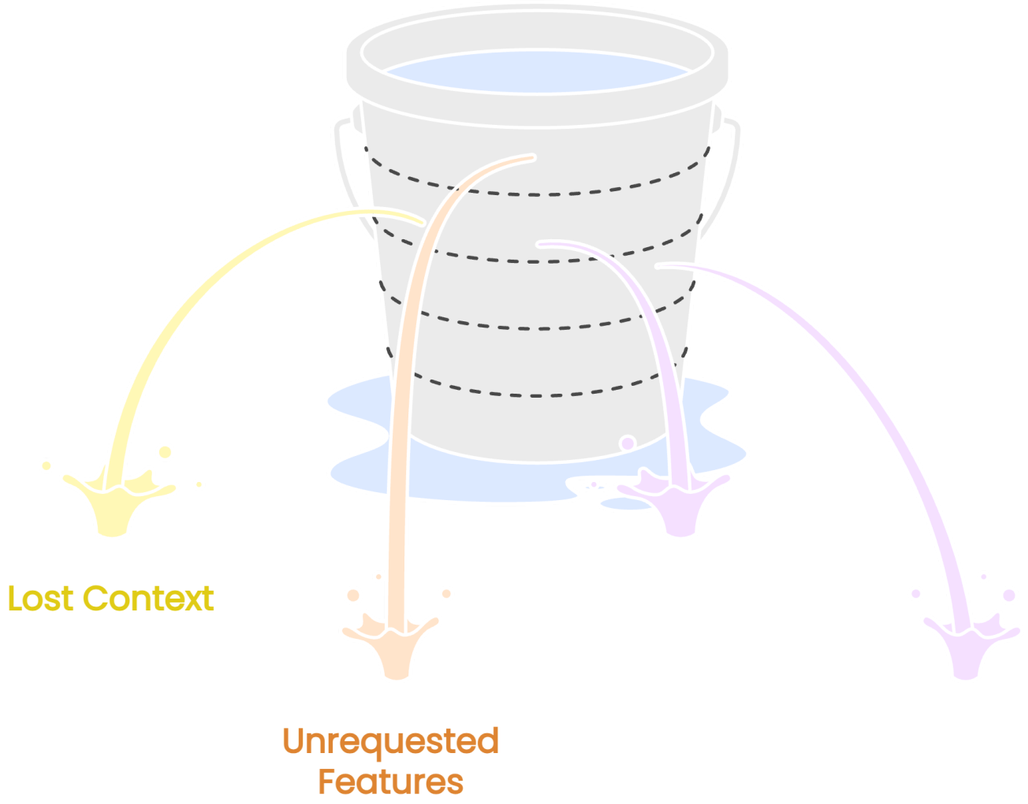


Vague prompt

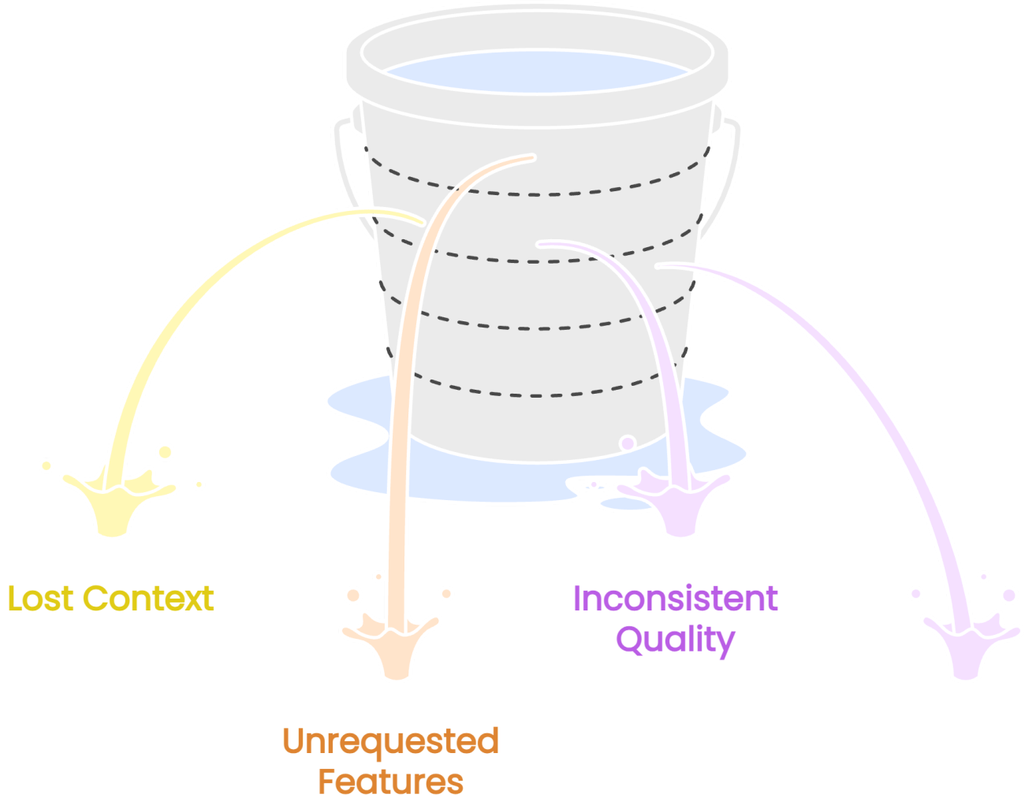
Vague prompts, weak results



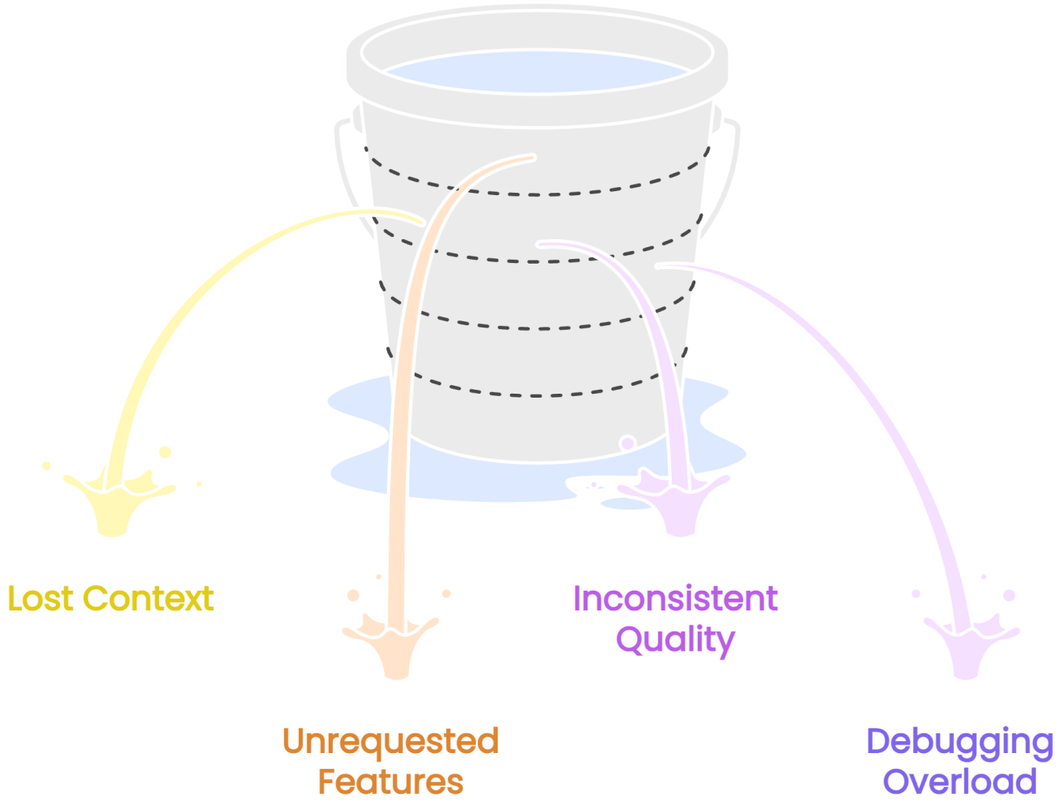
Vague prompts, weak results

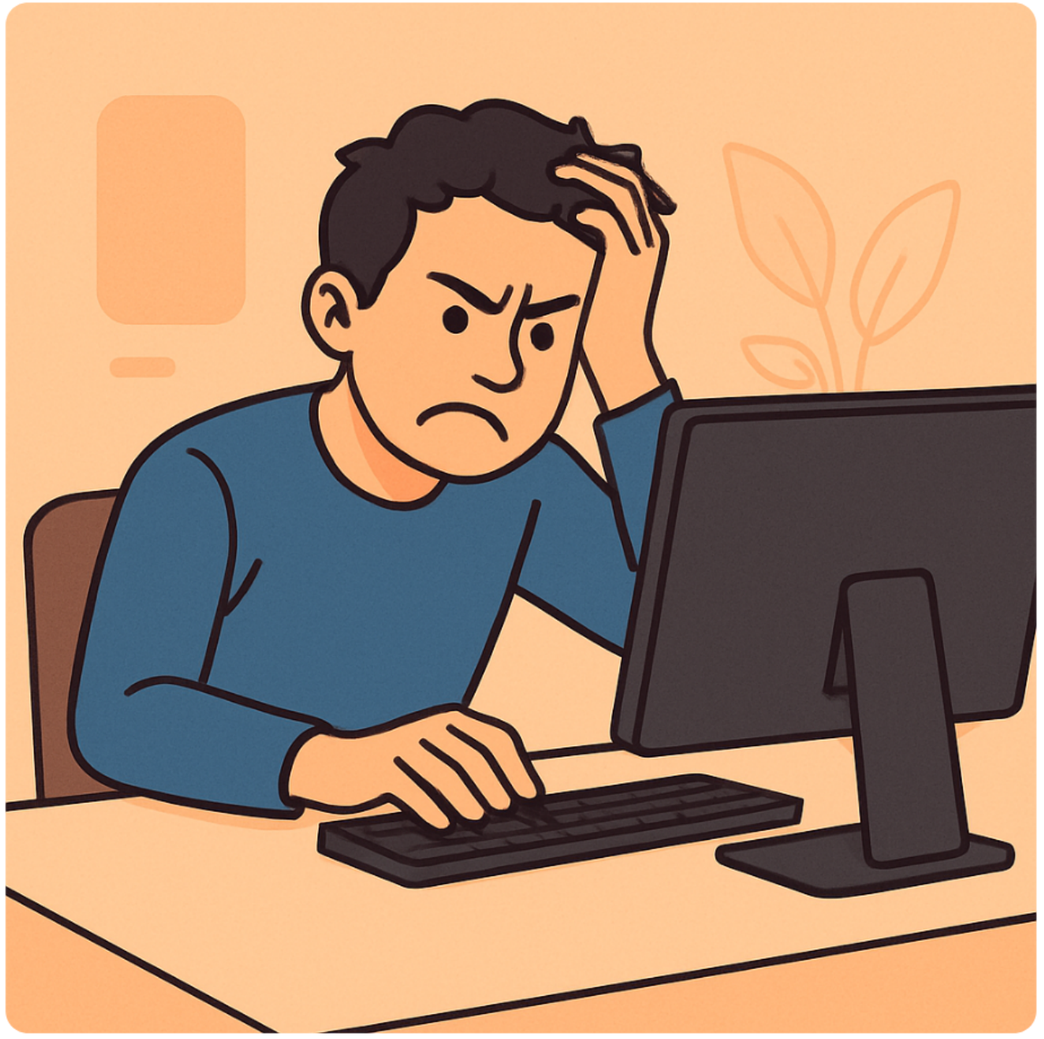


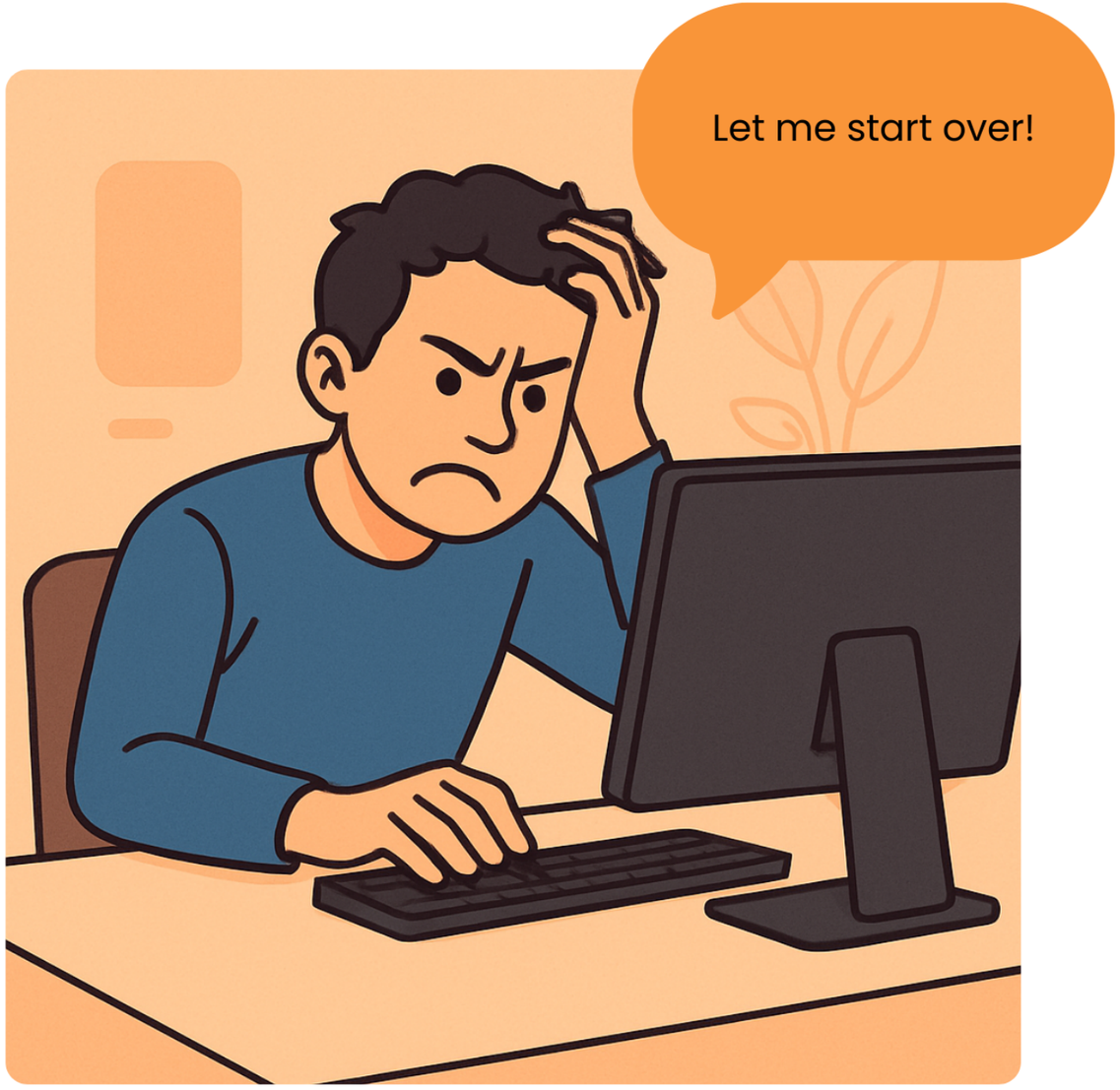
Vague prompts, weak results

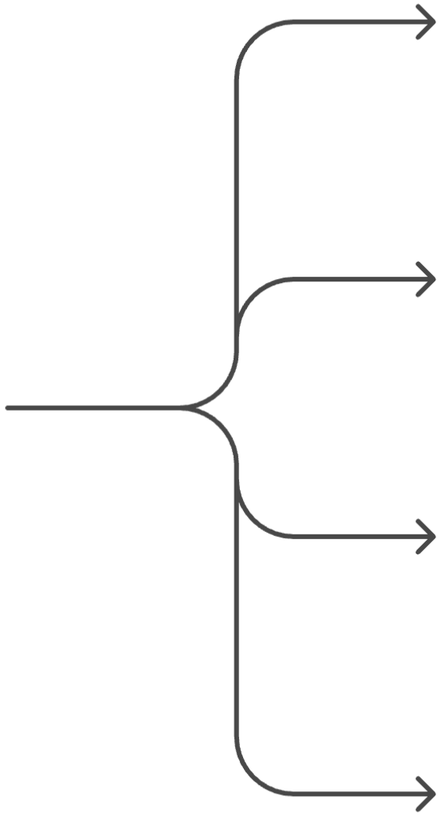


Vague prompts, weak results

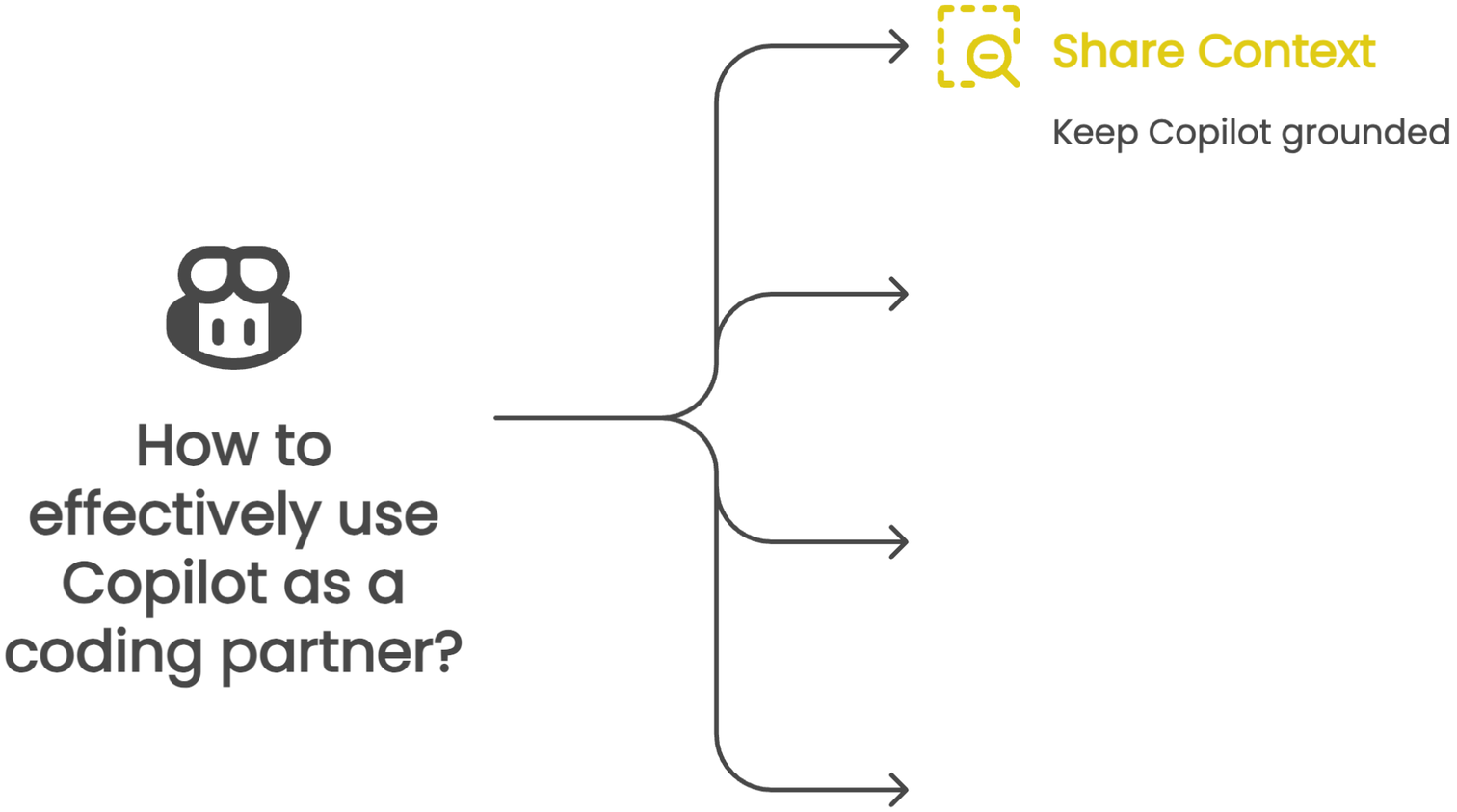


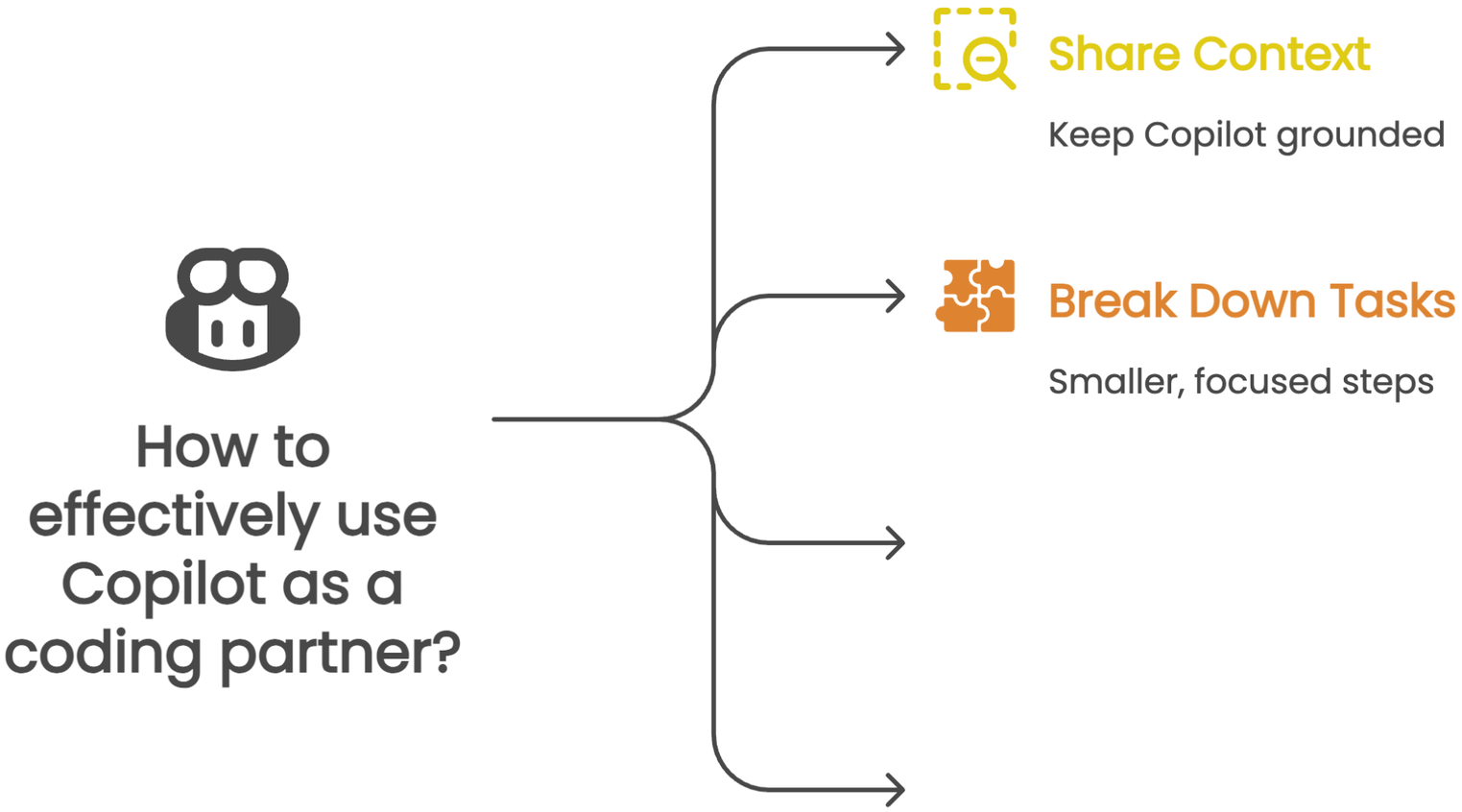



Let me start over!

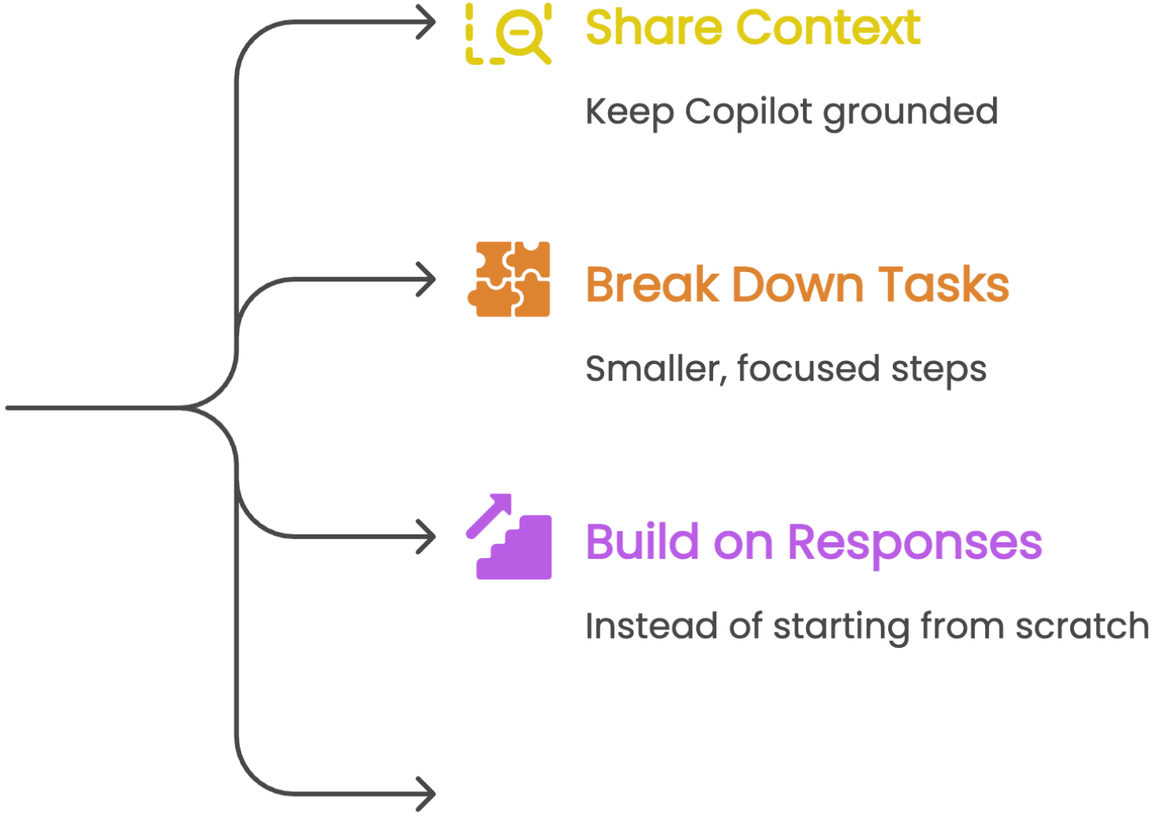


How to effectively use Copilot as a coding partner?


How to effectively use Copilot as a coding partner?
Share Context
Keep Copilot grounded


How to effectively use Copilot as a coding partner?
Share Context
Keep Copilot grounded
Break Down Tasks
Smaller, focused steps

How to effectively use Copilot as a coding partner?



Share Context

Keep Copilot grounded

Break Down Tasks

Smaller, focused steps

Build on Responses

Instead of starting from scratch

How to effectively use Copilot as a coding partner?

Share Context

Keep Copilot grounded

Break Down Tasks

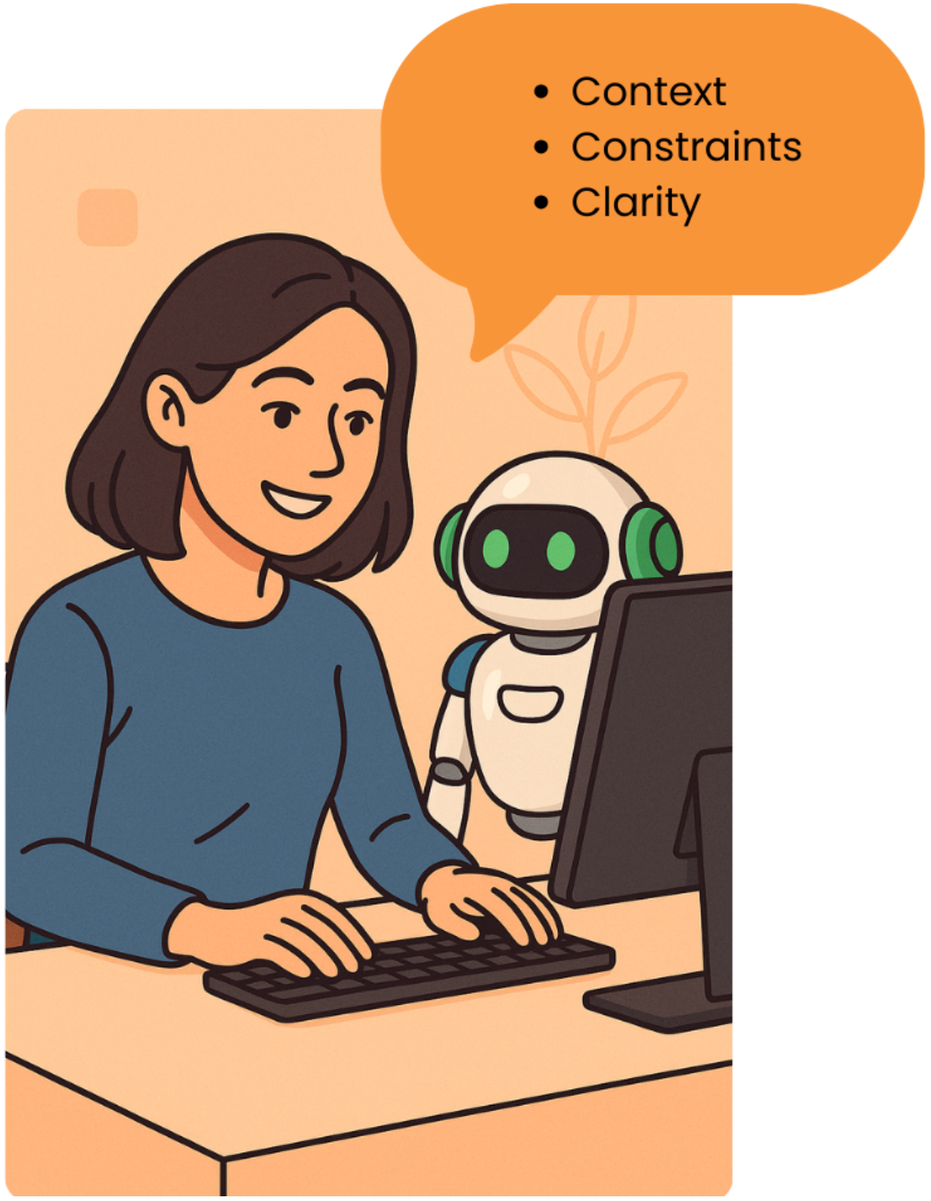
Smaller, focused steps

Build on Responses

Instead of starting from scratch

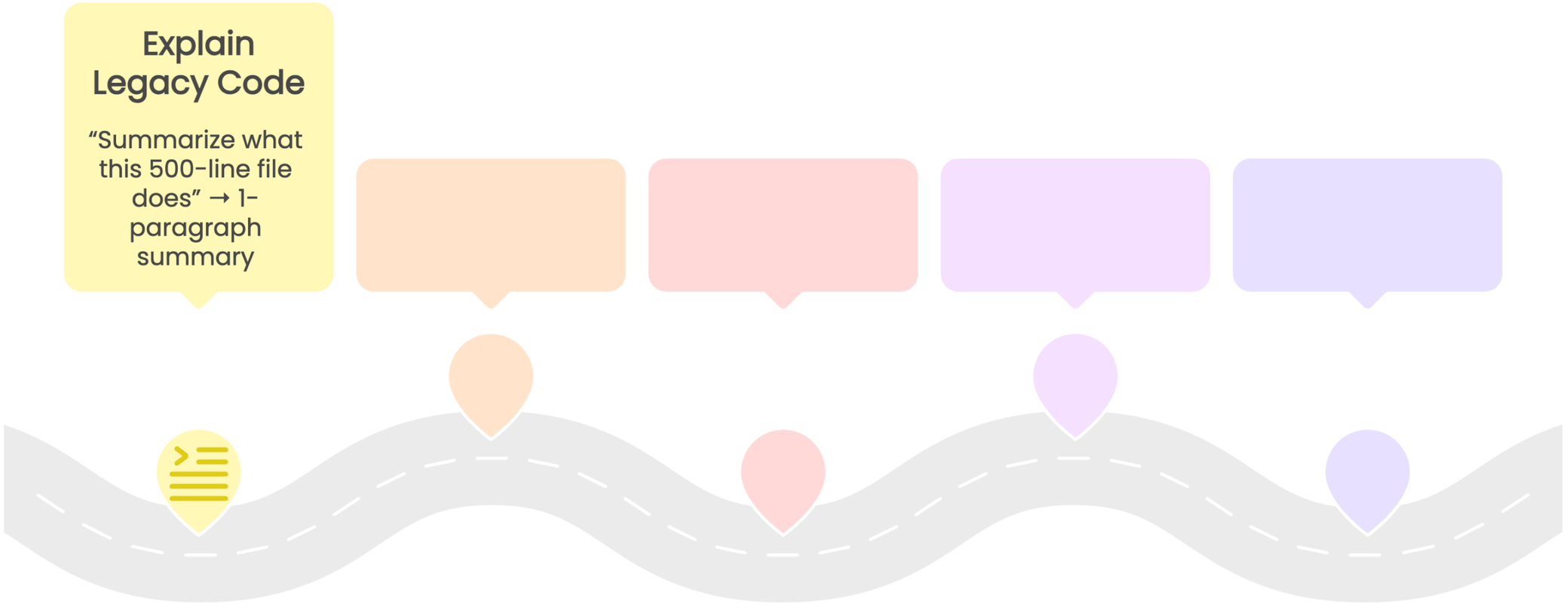
Choose Right Tools

Match the task at hand

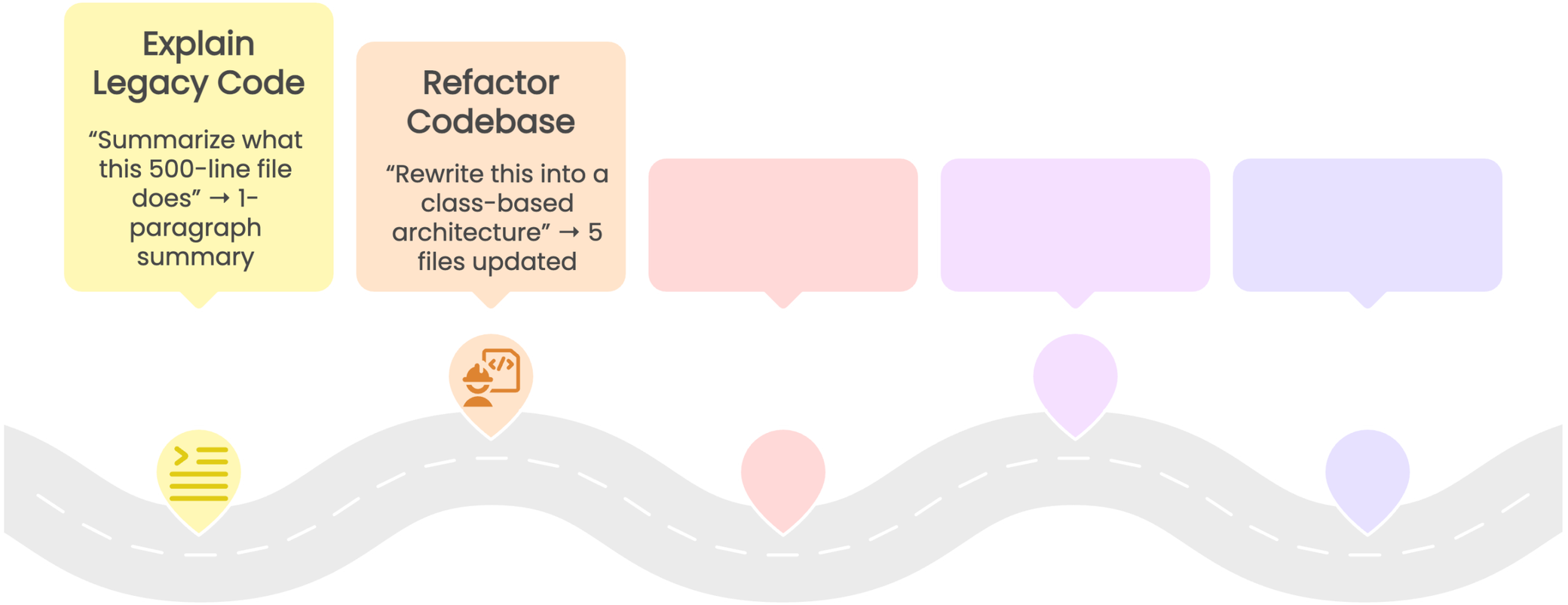

• Context
• Constraints
• Clarity

What can GitHub Copilot do?

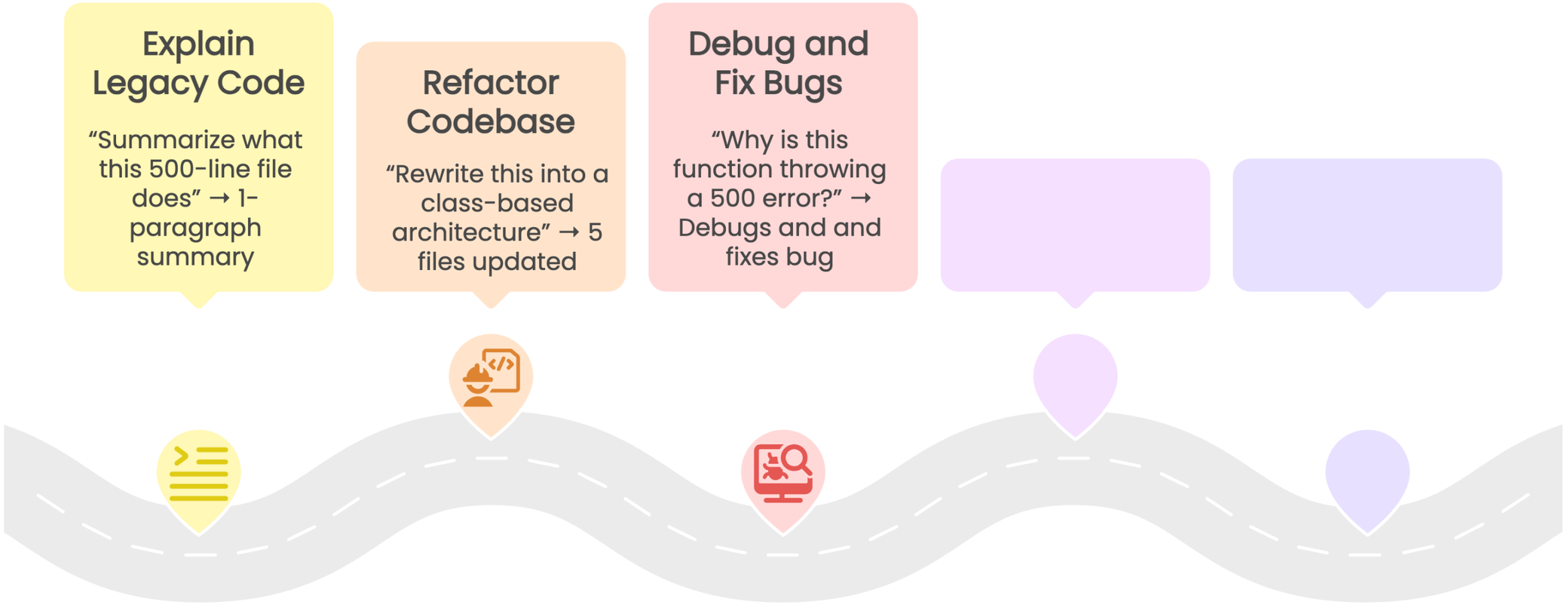
What can GitHub Copilot do?



What can GitHub Copilot do?



What can GitHub Copilot do?



What can GitHub Copilot do?

Explain Legacy Code

"Summarize what this 500-line file does" → 1-paragraph summary

Refactor Codebase

"Rewrite this into a class-based architecture" → 5 files updated

Debug and Fix Bugs

"Why is this function throwing a 500 error?" → Debugs and and fixes bug

Write Test Suites

"Write Pytest cases for my Flask app" → Writes 10 tests

What can GitHub Copilot do?

Explain Legacy Code

"Summarize what this 500-line file does" → 1-paragraph summary

Refactor Codebase

"Rewrite this into a class-based architecture" → 5 files updated

Debug and Fix Bugs

"Why is this function throwing a 500 error?" → Debugs and and fixes bug

Write Test Suites

"Write Pytest cases for my Flask app" → Writes 10 tests

Deployment Workflows

"Generate a Dockerfile and GitHub Actions pipeline" → YAML + Dockerfile

Let's practice!

SOFTWARE DEVELOPMENT WITH GITHUB COPILOT

Exploring GitHub Copilot modes

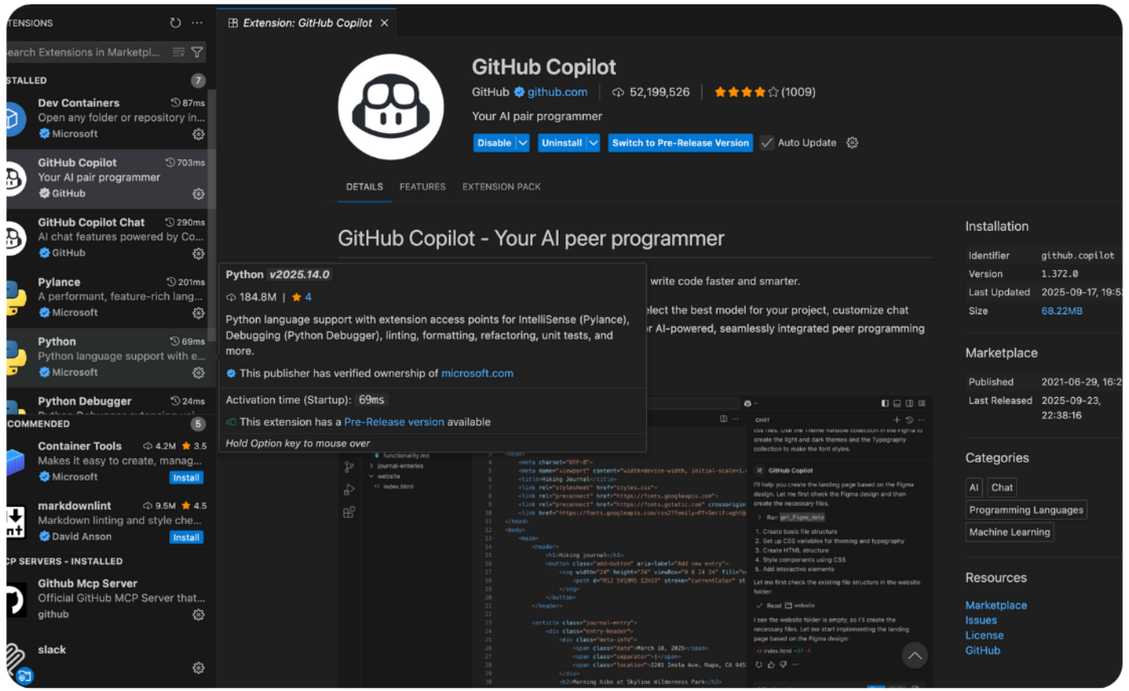
SOFTWARE DEVELOPMENT WITH GITHUB COPILOT

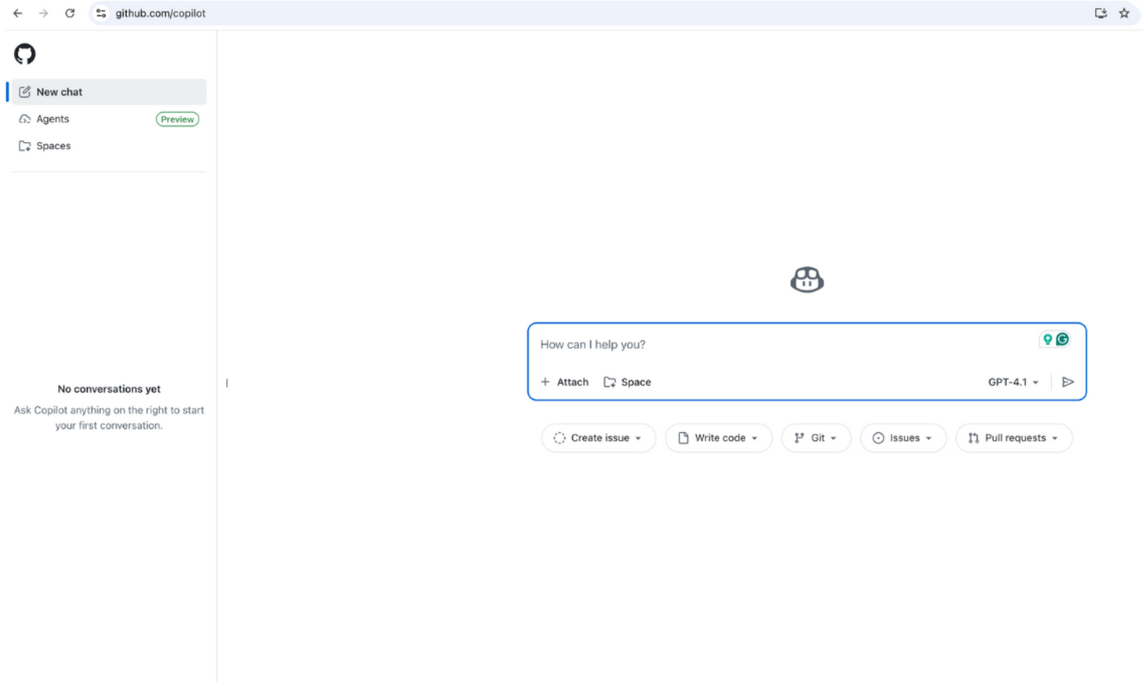
Thalia Barrera

AI Engineering Curriculum Manager, DataCamp

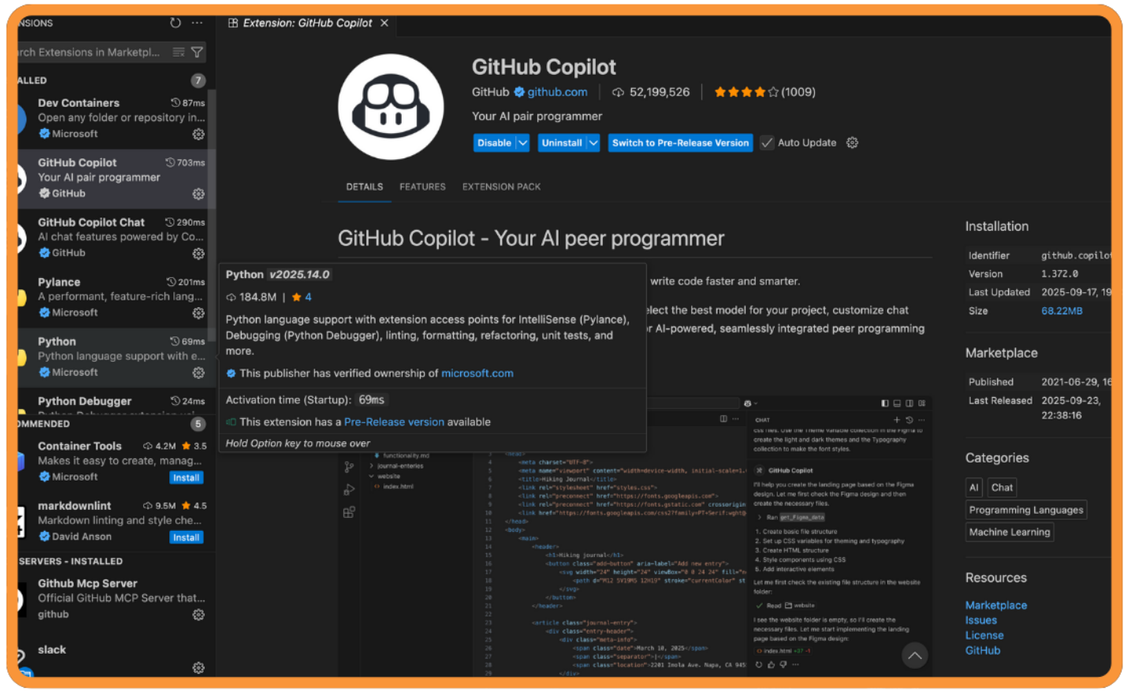
datacamp

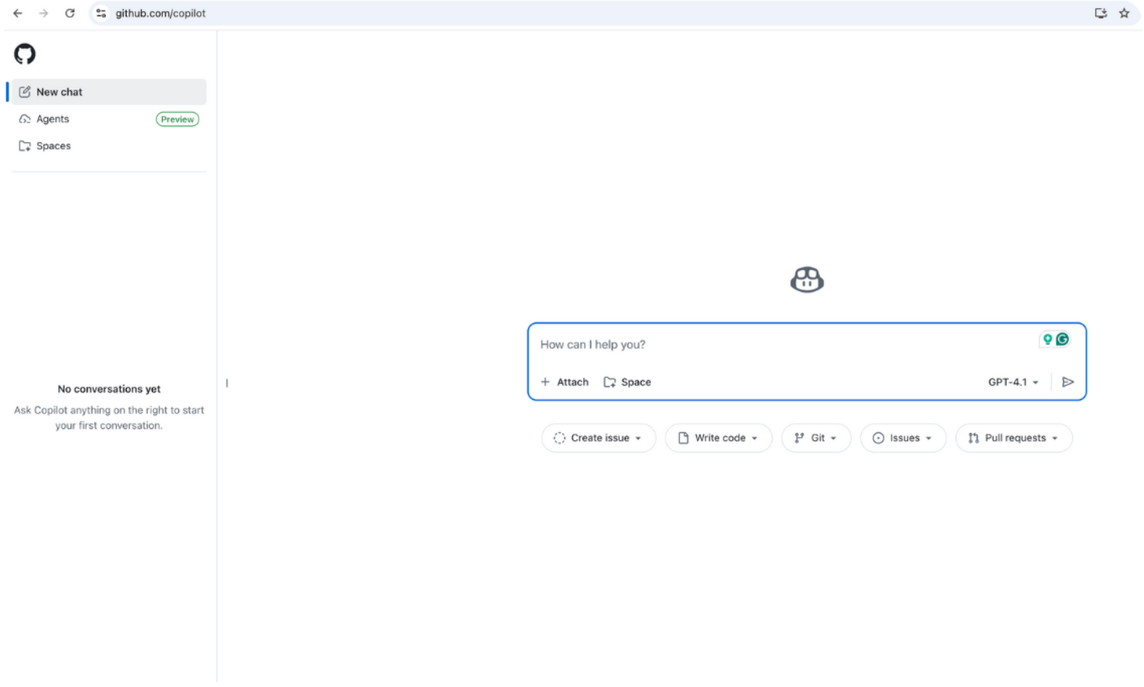
How to access GitHub Copilot

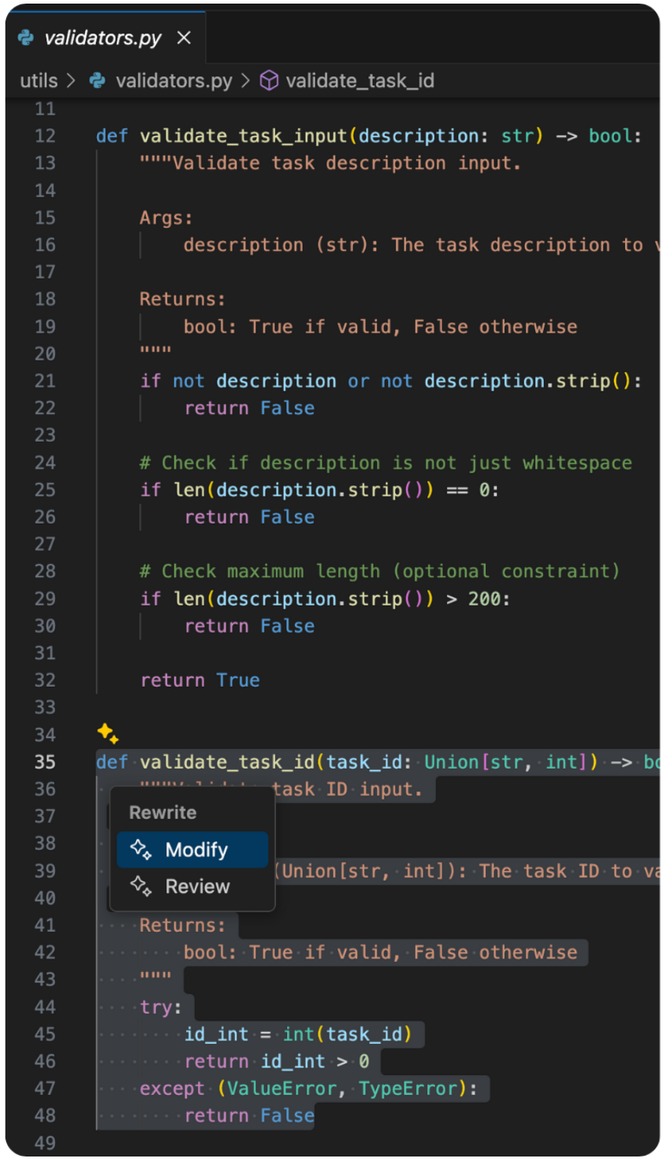


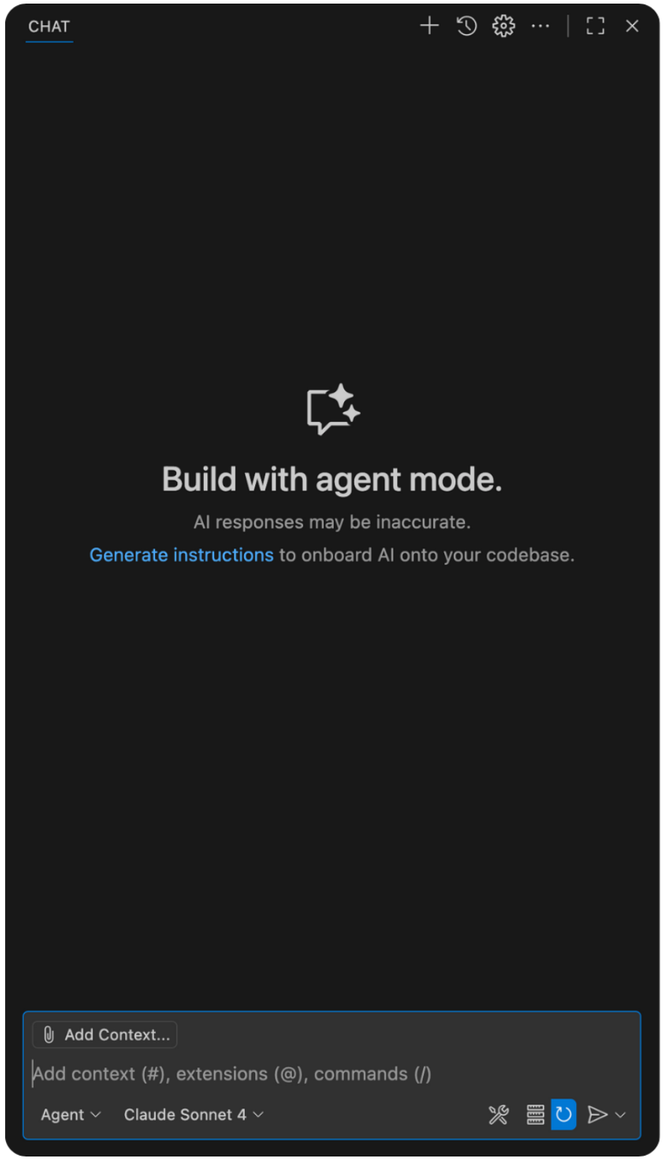


How to access GitHub Copilot

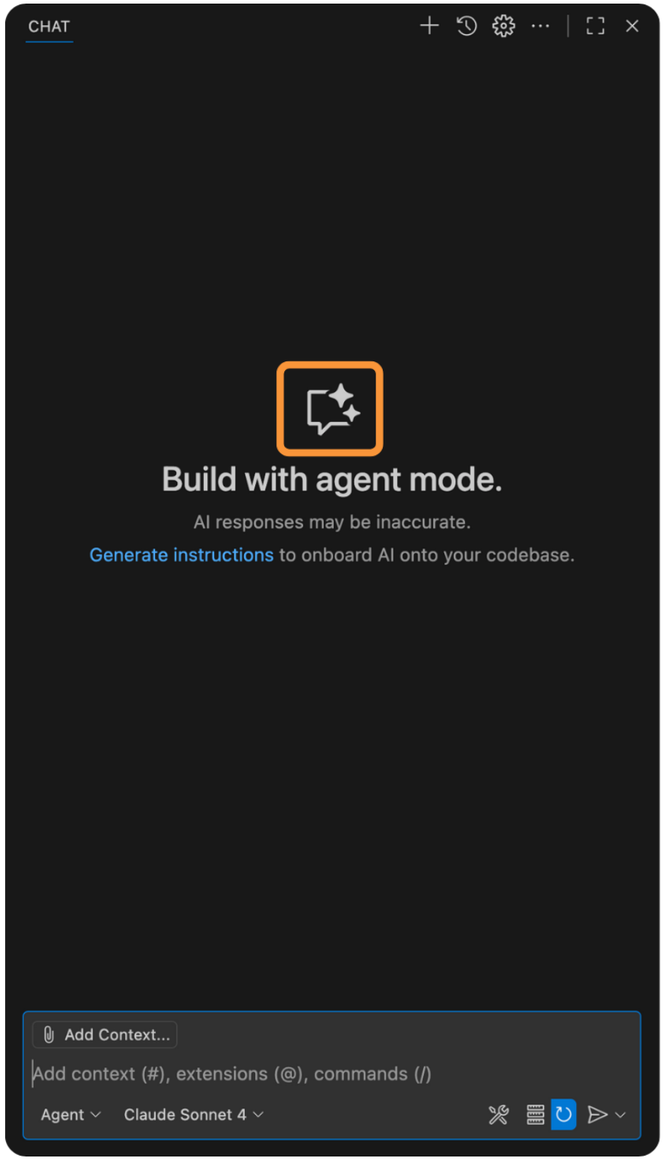





validators.py
utils > validators.py > validate_task_id
def validate_task_input(description: str) -> bool:
"""Validate task description input.
Args:
description (str): The task description to
Returns:
bool: True if valid, False otherwise
"""
if not description or not description.strip():
return False
# Check if description is not just whitespace
if len(description.strip()) == 0:
return False
# Check maximum length (optional constraint)
if len(description.strip()) > 200:
return False
return True
def validate_task_id(task_id: Union[str, int]) -> bo
task ID input.
(Union[str, int]): The task ID to va
Returns:
bool: True if valid, False otherwise
"""
try:
id_int = int(task_id)
return id_int > 0
except (ValueError, TypeError):
return False
Rewrite
Modify
Review


CHAT
Build with agent mode.
AI responses may be inaccurate.
Generate instructions to onboard AI onto your codebase.
Add Context...
Add context (#), extensions (@), commands (/)
Agent
Claude Sonnet 4


validators.py
utils > validators.py > validate_task_id
def validate_task_input(description: str) -> bool:
"""Validate task description input.
Args:
description (str): The task description to
Returns:
bool: True if valid, False otherwise
"""
if not description or not description.strip():
return False
# Check if description is not just whitespace
if len(description.strip()) == 0:
return False
# Check maximum length (optional constraint)
if len(description.strip()) > 200:
return False
return True
def validate_task_id(task_id: Union[str, int]) -> bo
task ID input.
Rewrite
Modify
Review
(Union[str, int]): The task ID to va
Returns:
bool: True if valid, False otherwise
"""
try:
id_int = int(task_id)
return id_int > 0
except (ValueError, TypeError):
return False


CHAT
Build with agent mode.
AI responses may be inaccurate.
Generate instructions to onboard AI onto your codebase.
Add Context...
Add context (#), extensions (@), commands (/)
Agent
Claude Sonnet 4

When to use Autocomplete or Inline Chat

- Quick code modifications
- Adding missing logic
- Fixing small bugs
- Refactoring a specific section

When to use Ask Mode

- Plan your application's architecture
- Explore approaches to a particular problem
- Walk through implementation steps
- Reference multiple files and concepts

When to use Edit Mode

- Multi-file refactoring
- Feature implementation involving multiple components
- Setting up new project structures
- Complex operations that require understanding relationships between files

Autocomplete or Inline Chat

Speed and focus

Autocomplete or Inline Chat

Speed and focus

Ask Mode

Thinking and planning

Autocomplete or Inline Chat

Speed and focus

Ask Mode

Thinking and planning

Edit Mode

Complex tasks

Let's practice!

SOFTWARE DEVELOPMENT WITH GITHUB COPILOT

Introducing Agent Mode

SOFTWARE DEVELOPMENT WITH GITHUB COPILOT

Thalia Barrera
AI Engineering Curriculum Manager, DataCamp

datacamp

How is Agent Mode different?

How is Agent Mode different?

It's designed for complex, multi-step tasks.

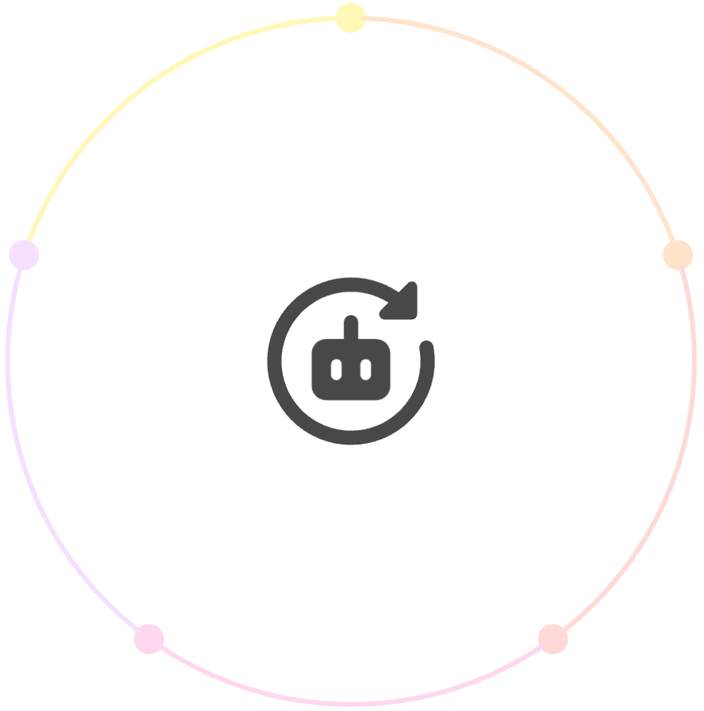
How is Agent Mode different?

It's designed for complex, multi-step tasks.

It works iteratively and autonomously to achieve a goal.

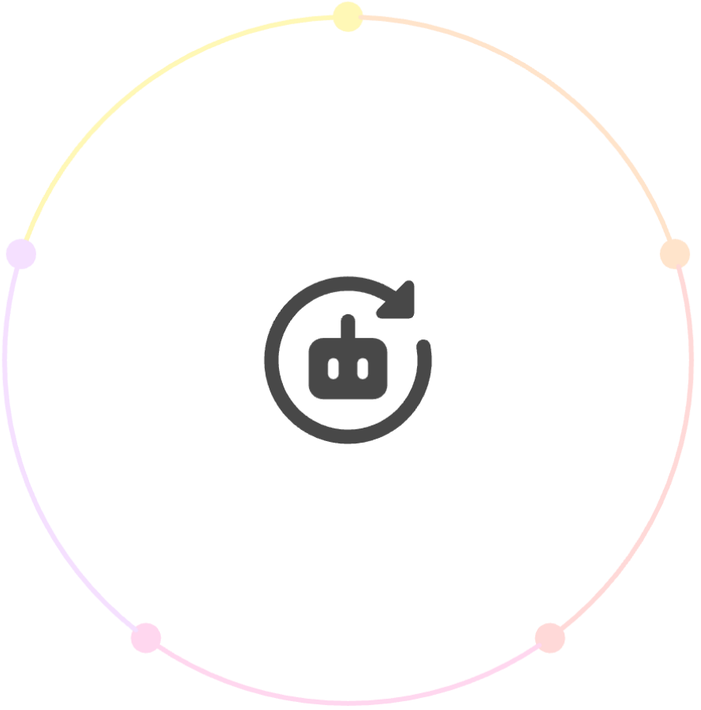
Suggest Solution

Offers an initial solution



Suggest Solution

Offers an initial solution

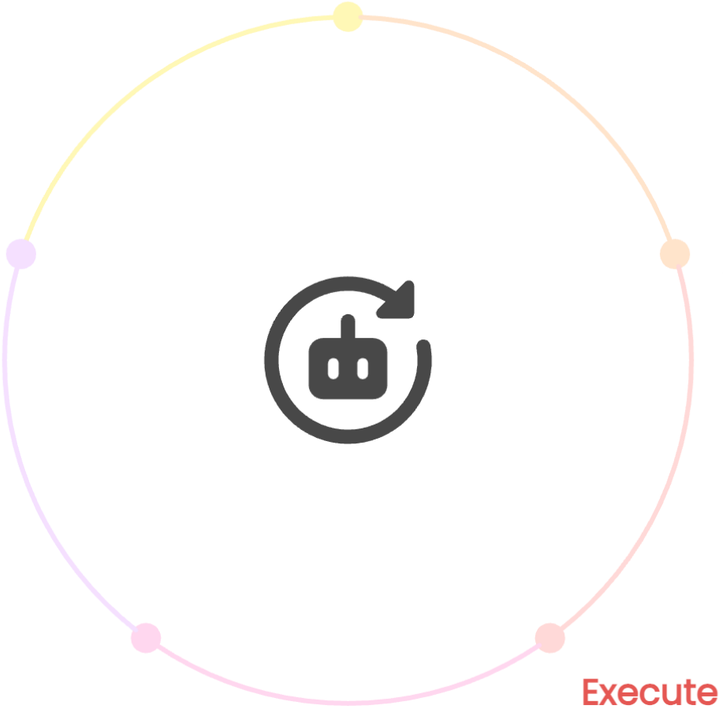


Implement Changes

Makes changes across files

Suggest Solution

Offers an initial solution



Implement Changes

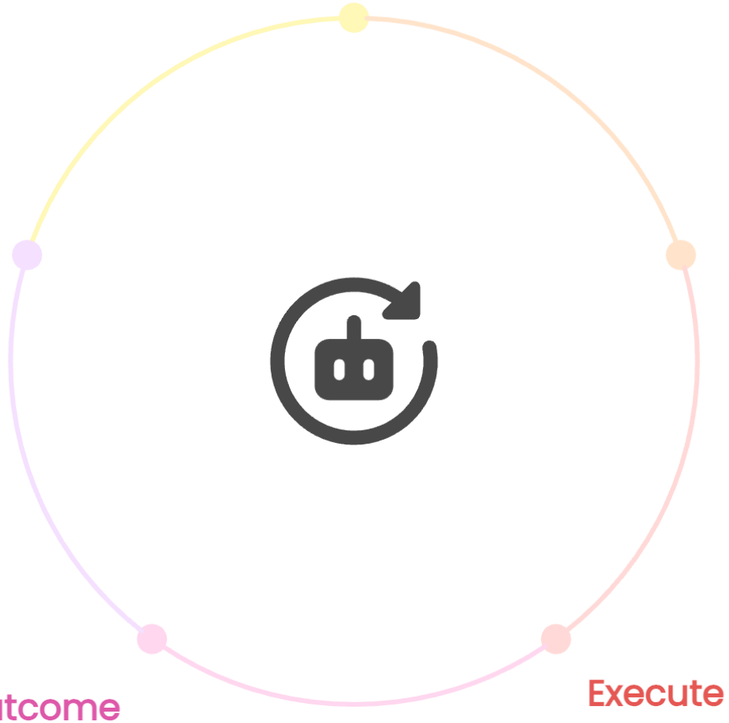
Makes changes across files

Execute Tools

Runs necessary tools or commands

Suggest Solution

Offers an initial solution



Implement Changes

Makes changes across files

Execute Tools

Runs necessary tools or commands

Evaluate Outcome

Assesses the effectiveness of the solution

Suggest Solution

Offers an initial solution

Implement Changes

Makes changes across files

Execute Tools

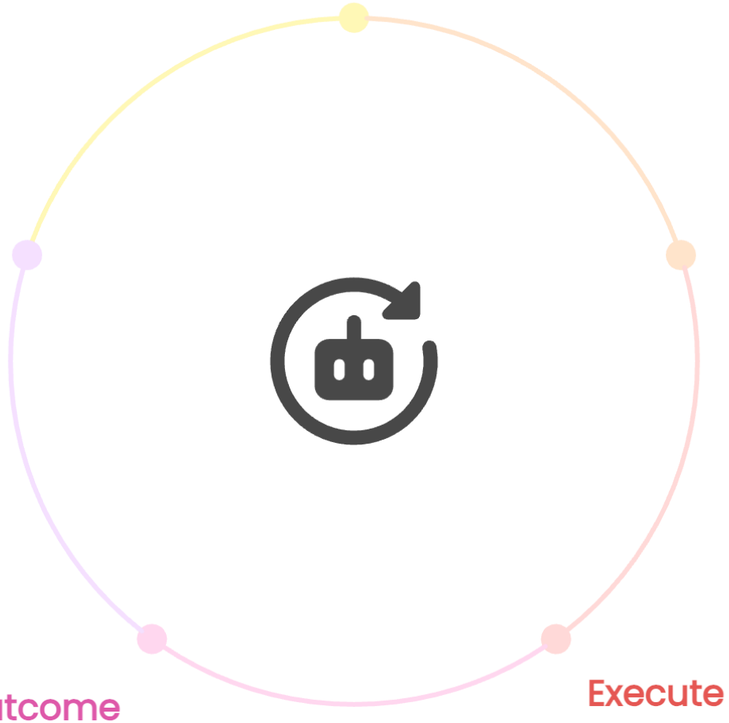
Runs necessary tools or commands

Evaluate Outcome

Assesses the effectiveness of the solution

Iterate if Needed

Tries alternative solutions if the first fails

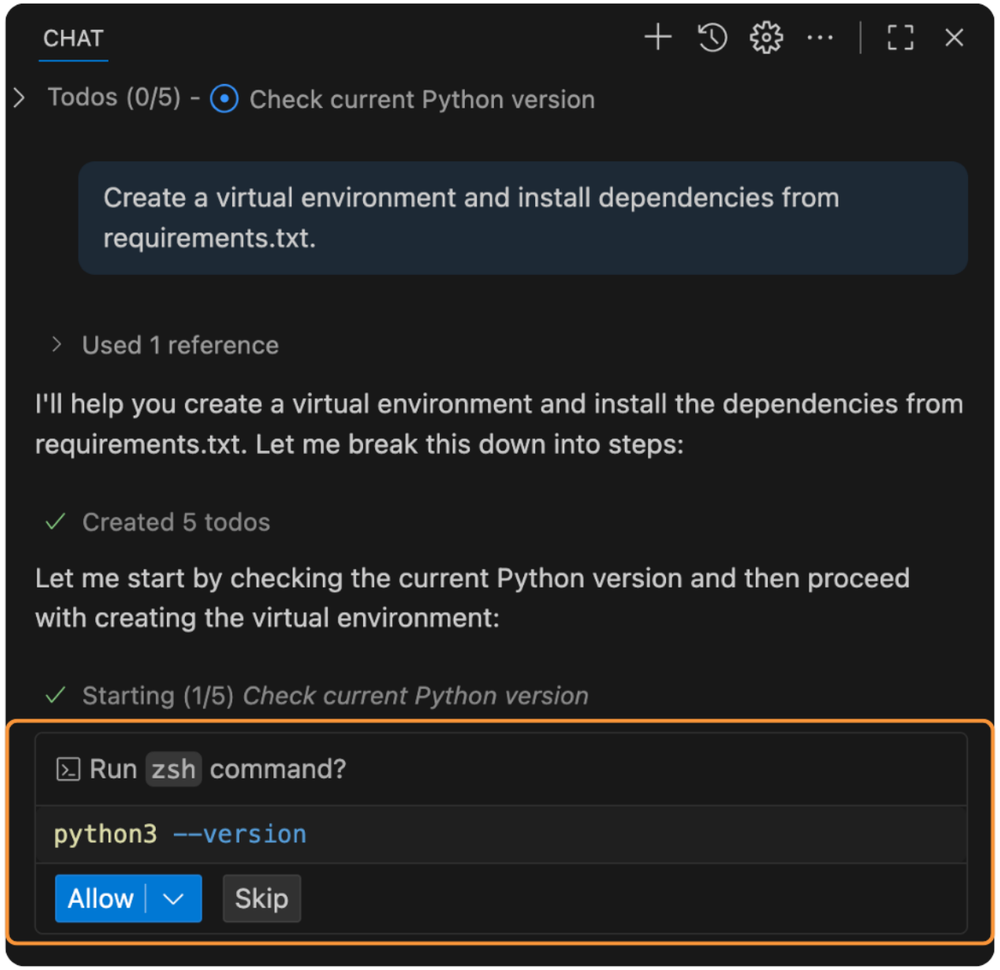


When to use Agent Mode

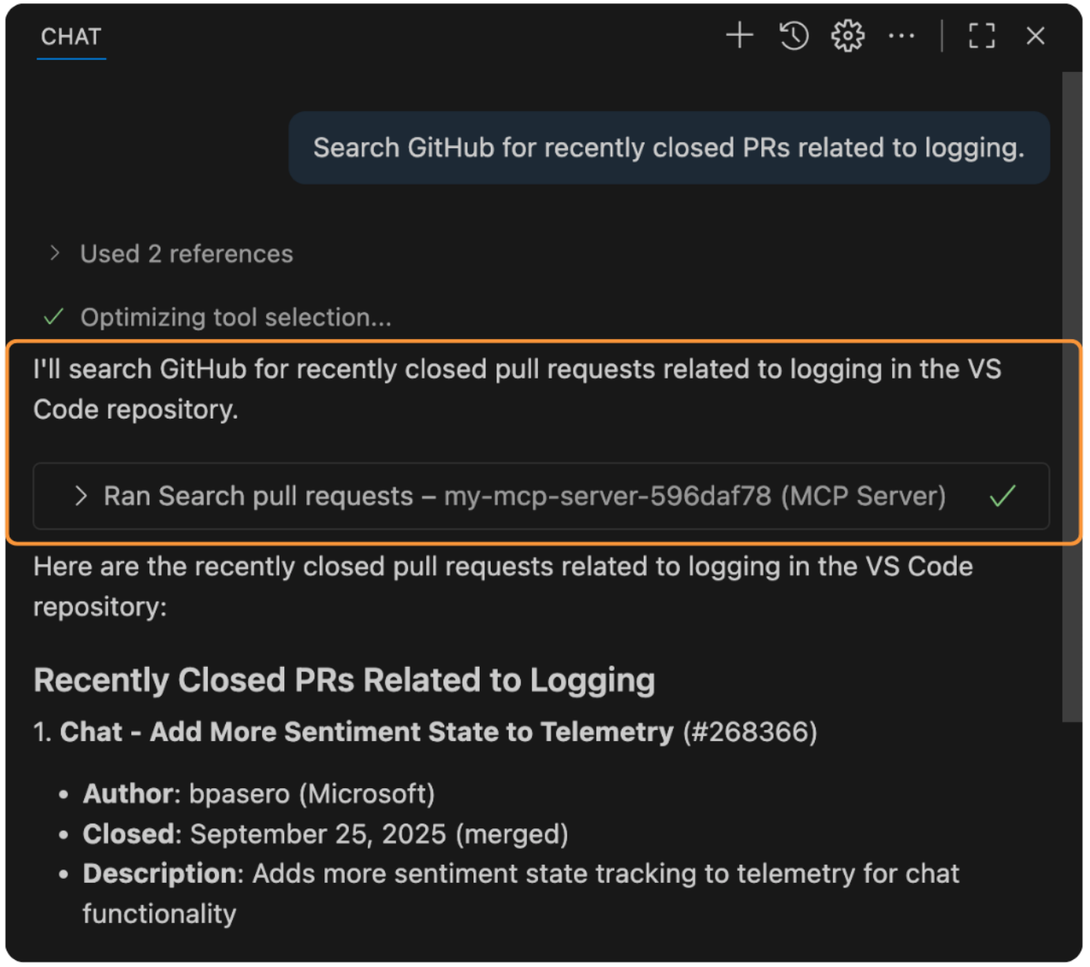
- Task spans multiple files
- Requires code edits and command execution
- Involves experimentation, refactoring, or retries


How did you do that?
I used a tool!

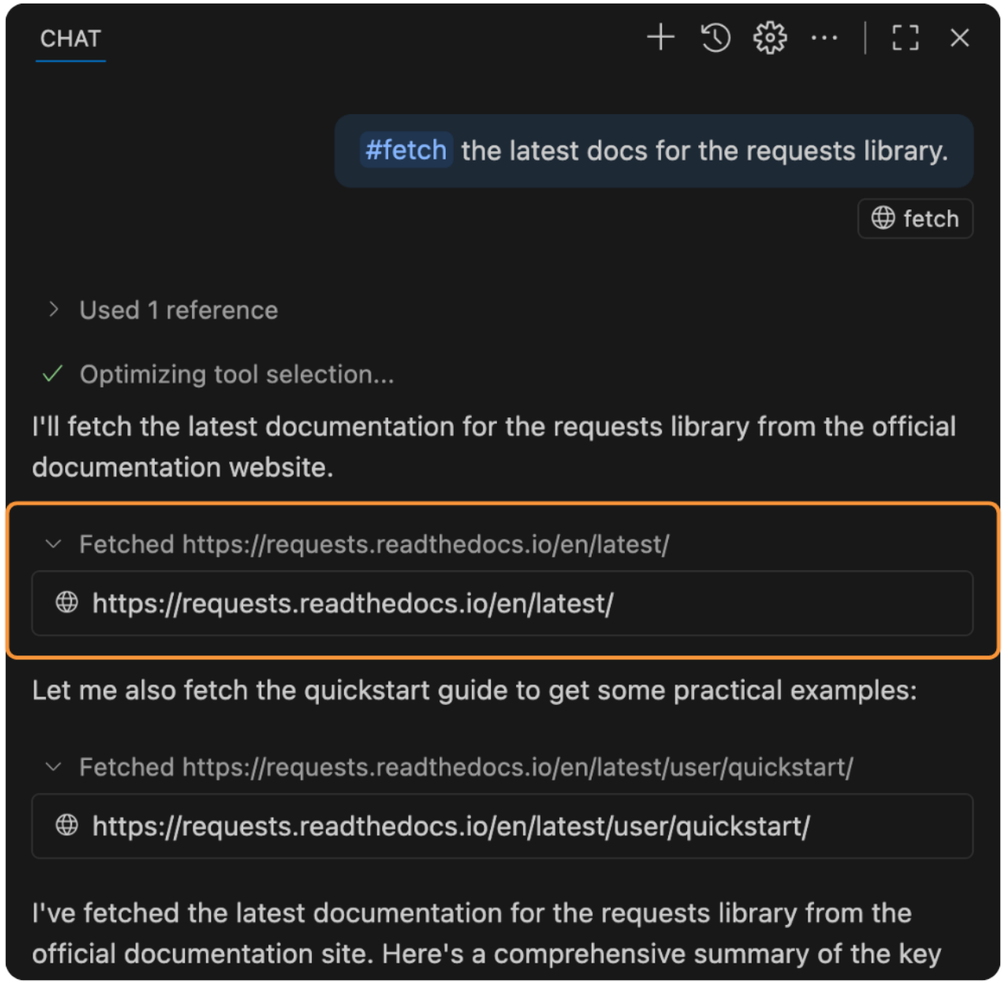
Agent Mode tool usage examples



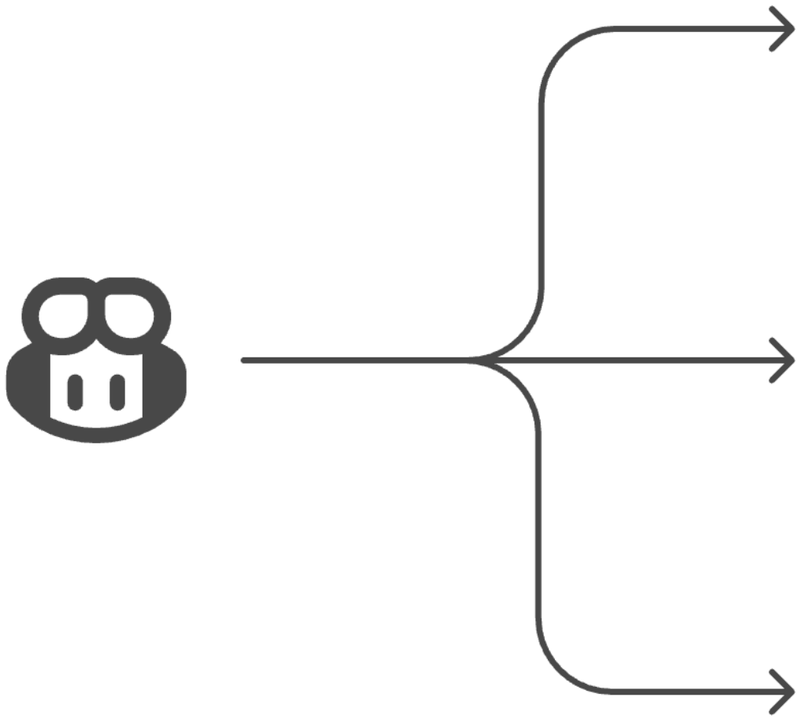
Agent Mode tool usage examples



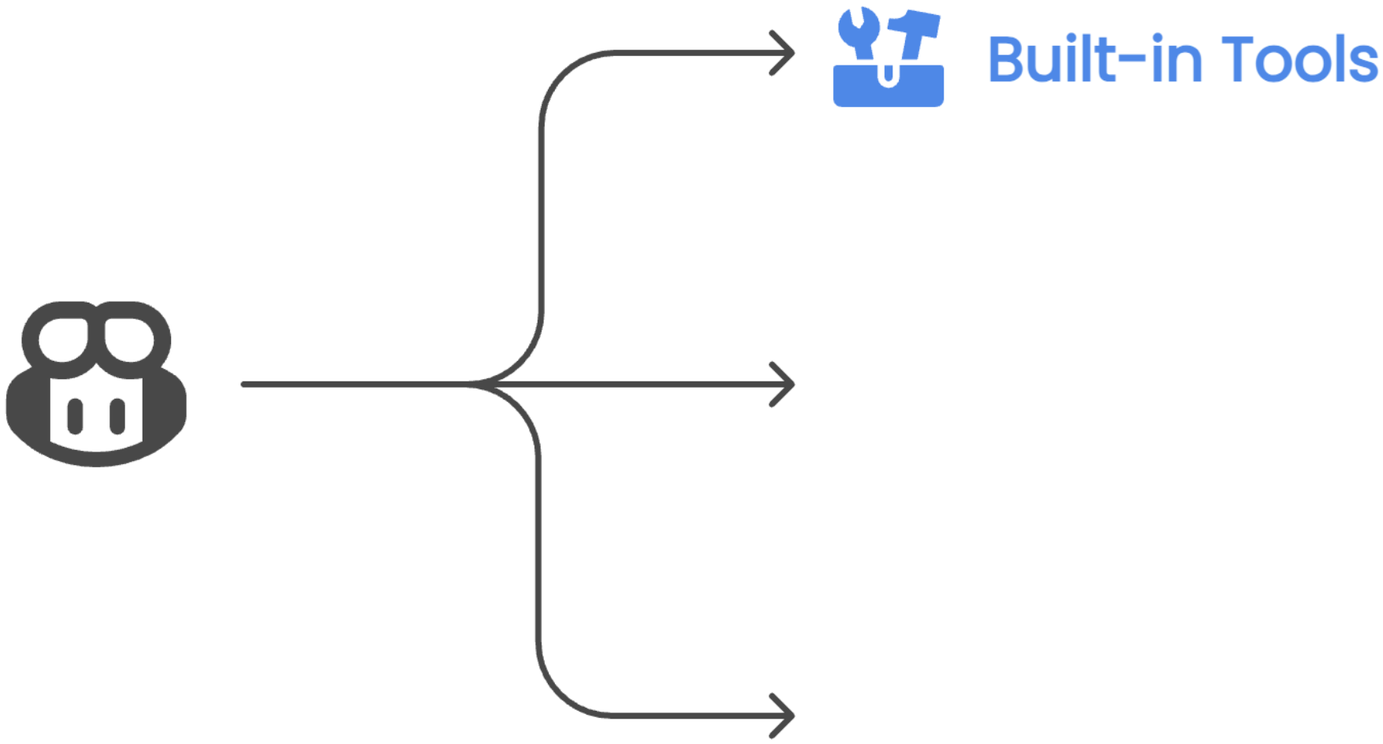
Agent Mode tool usage examples



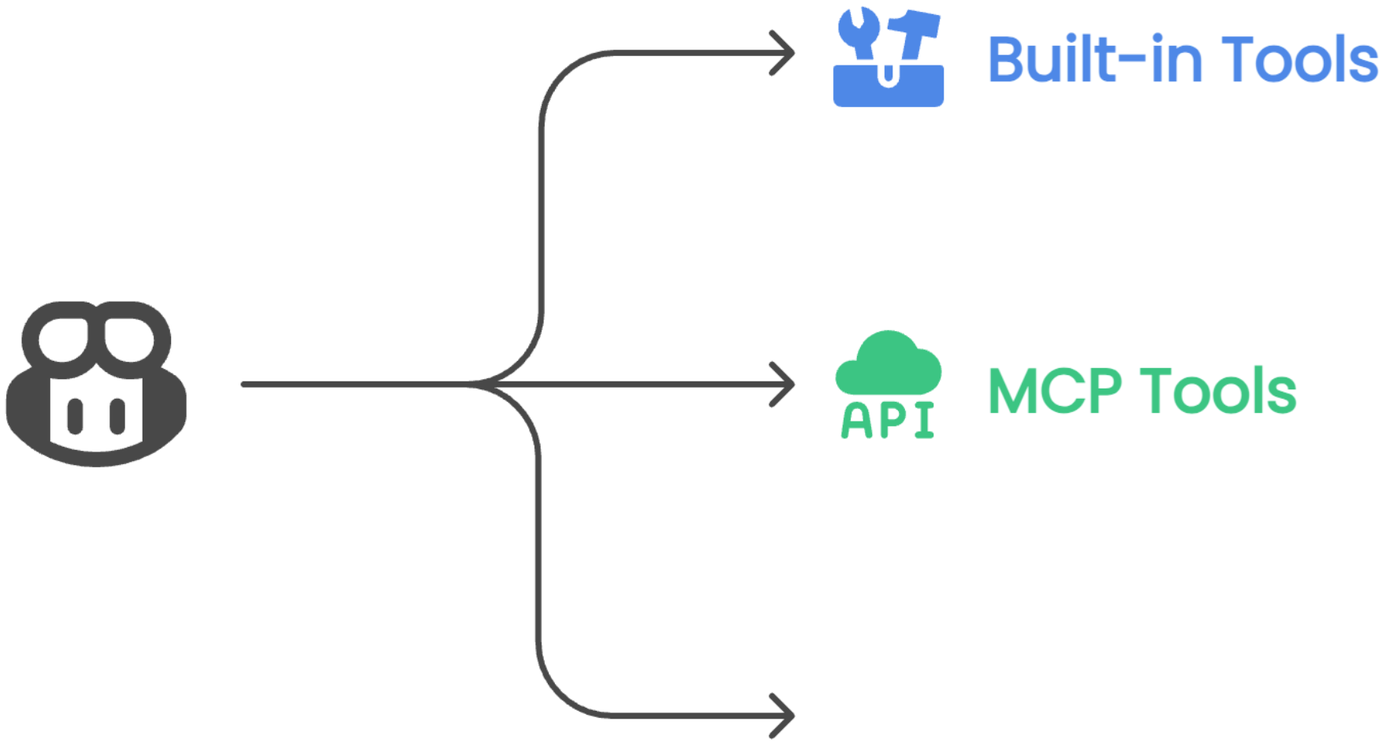
Types of tools Copilot uses



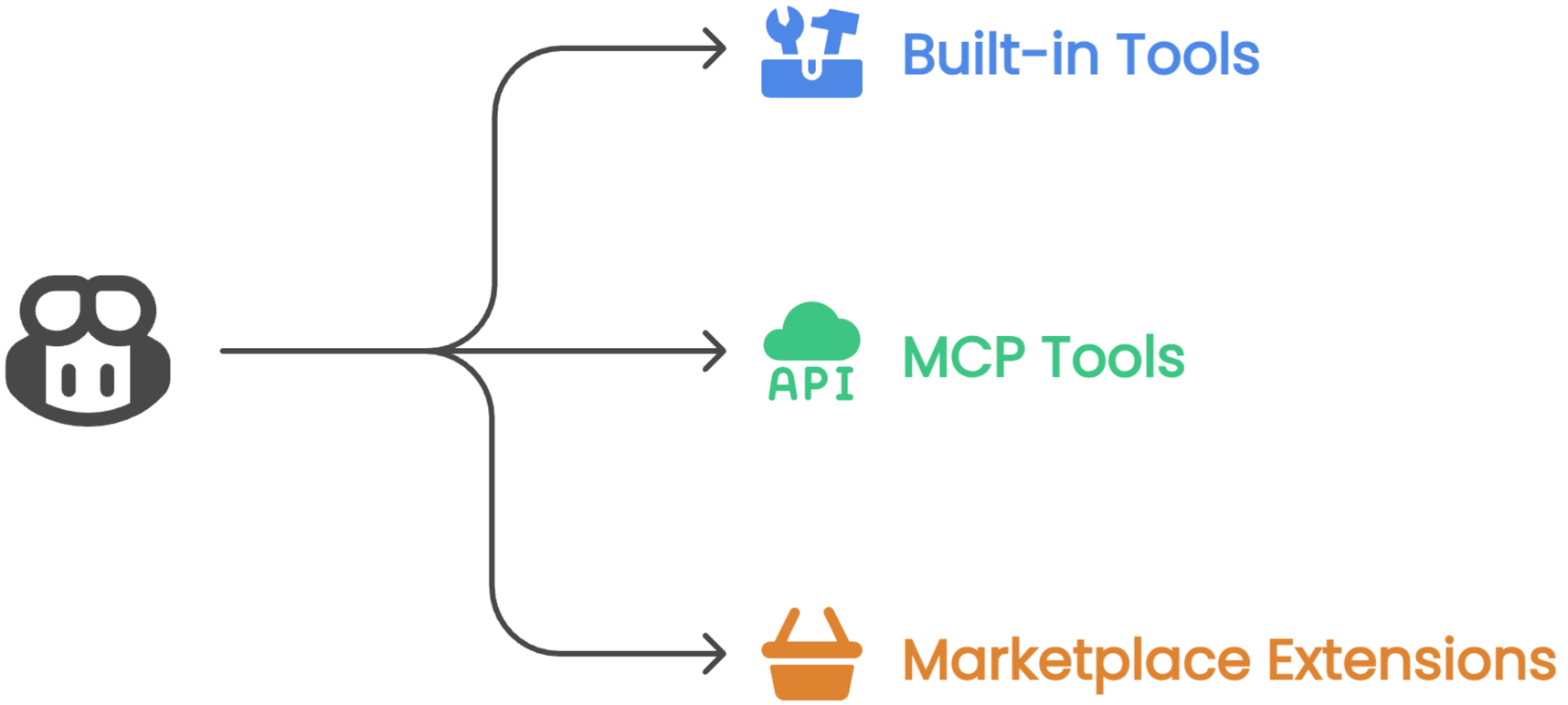
Types of tools Copilot uses



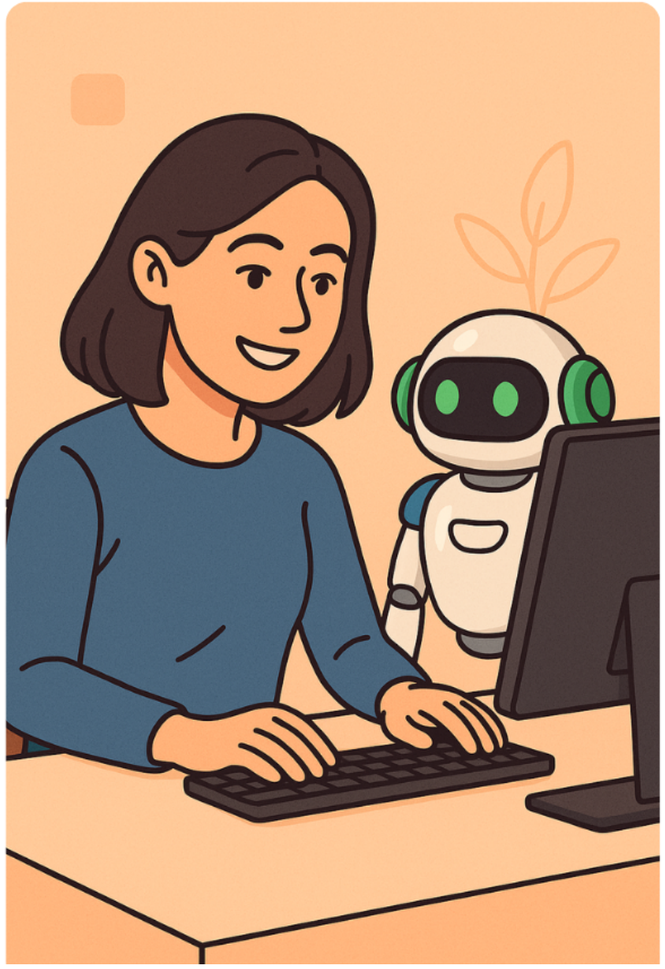
Types of tools Copilot uses



Types of tools Copilot uses



Agent Mode: coding assistant



Agent Mode: coding assistant

Let's practice!

SOFTWARE DEVELOPMENT WITH GITHUB COPILOT