

IDE ❤️ Agents

An opinionated guide to AI coding in 2025



Silas Alberti

Founding Team @ Cognition
Prev: Stanford PhD Student

W Windsurf



Devin

Agenda

1. Overview of the AI tooling landscape
2. Synchronous vs. asynchronous tools
3. The 2025 coding workflow
 - When to hand-off from sync to async?
 - How to combine tools like Devin & Windsurf
4. Where are we headed?

Three Eras of AI Coding Tools

Local Development ⇒ Collaborative Cloud Agents

1

GitHub Copilot: speed up coding

Code Completion

```
> print Greeting
```

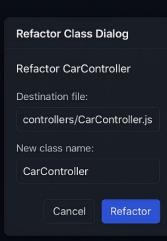
```
1 def greet:  
2     # Print a greeting  
3     print()
```

2

AI IDEs: single-player task completion

IDE Automation

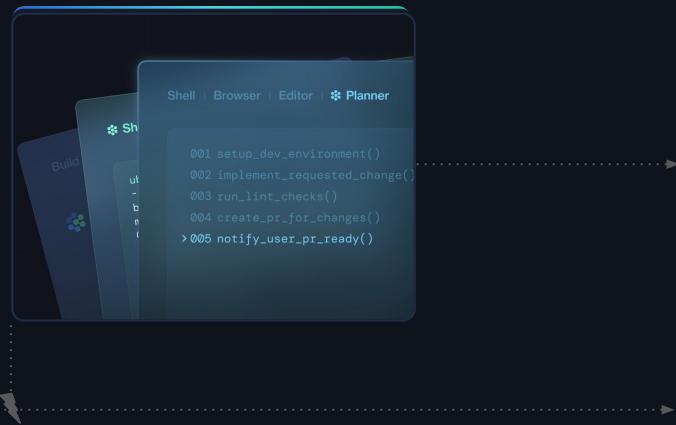
```
import express;  
import CarService from 'CarService';  
  
class CarController extends CarService {  
    constructor() {  
        super();  
    }  
  
    Refactor to suggested class...  
    async handleGetCar(req, res) {  
        try {  
            const carId = req.params.id;  
            const car = await CarController.findById(carId);  
            res.json(car);  
        } catch (err) {  
            res.status(500).json({  
                error: err.message,  
            });  
        }  
    }  
}
```



3

AI agents: scale workflows in parallel

AI Software Engineer



Three Eras of AI Coding Tools

Local Development ⇒ Collaborative Cloud Agents

1 ~10% efficiency gain

GitHub Copilot: speed up coding

Code Completion

```
> print Greeting
```

```
1 def greet:  
2     # Print a greeting  
3     print()
```

```
import express;
import CarService from 'CarService';

class CarController extends
    carService() {
    constructor: extends(C
    );
}

Refactor to suggested class...
async handleGetCar(req, res)
try {
    car EId = req.params;
    console CarController
```

AI IDEs: single-player task completion

IDE Automation

3

6-12x efficiency gain

AI agents: scale workflows in parallel

AI Software Engineer

Cloud, Asynchronous



Local, Synchronous



Synchronous vs. Asynchronous

sync:

single-threaded, human-in-the-loop, your attention is focused on one task
=> AI agent works for 20 seconds - 1.5 minutes

Synchronous vs. Asynchronous

sync:

single-threaded, human-in-the-loop, your attention is focused on one task

=> AI agent works for 20 seconds - 1.5 minutes

async:

multi-threaded, human delegates to AI, switches attention between multiple tasks

=> AI agent works for 10 minutes - multiple hours

Sync

Async

Local

Cloud

Sync

Async

Local

 Windsurf

 CURSOR

 GitHub Copilot

Cloud

Sync

Async

Local

 Windsurf

 CURSOR

 GitHub Copilot

Cloud

 Devin

 Codex

Local

Sync

Async

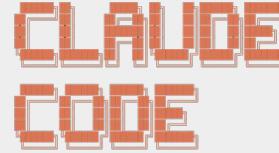
 Windsurf

 CURSOR

 GitHub Copilot

Cloud

DeepWiki

 CLAUDE
CODE

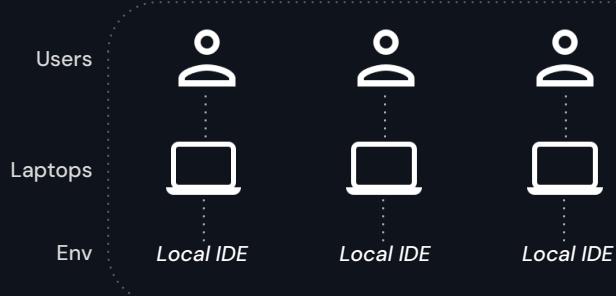
 Devin

 Codex

Cloud + Async enables 10x parallelism

Local AI IDEs

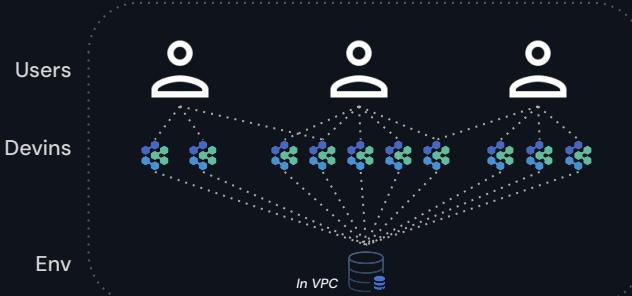
Boost individual speed.



Local • Synchronous • 1-to-1 • Isolated Knowledge

Cloud AI Agents

Unlimited Devins for parallel [capacity](#).



Cloud • Asynchronous • 1-to-Many • Organizational Knowledge

Using async agents is a hard but learnable skill

Managing async agents can unlock 10x gains...
...but most people use sync agents.

Using async agents is a hard but learnable skill

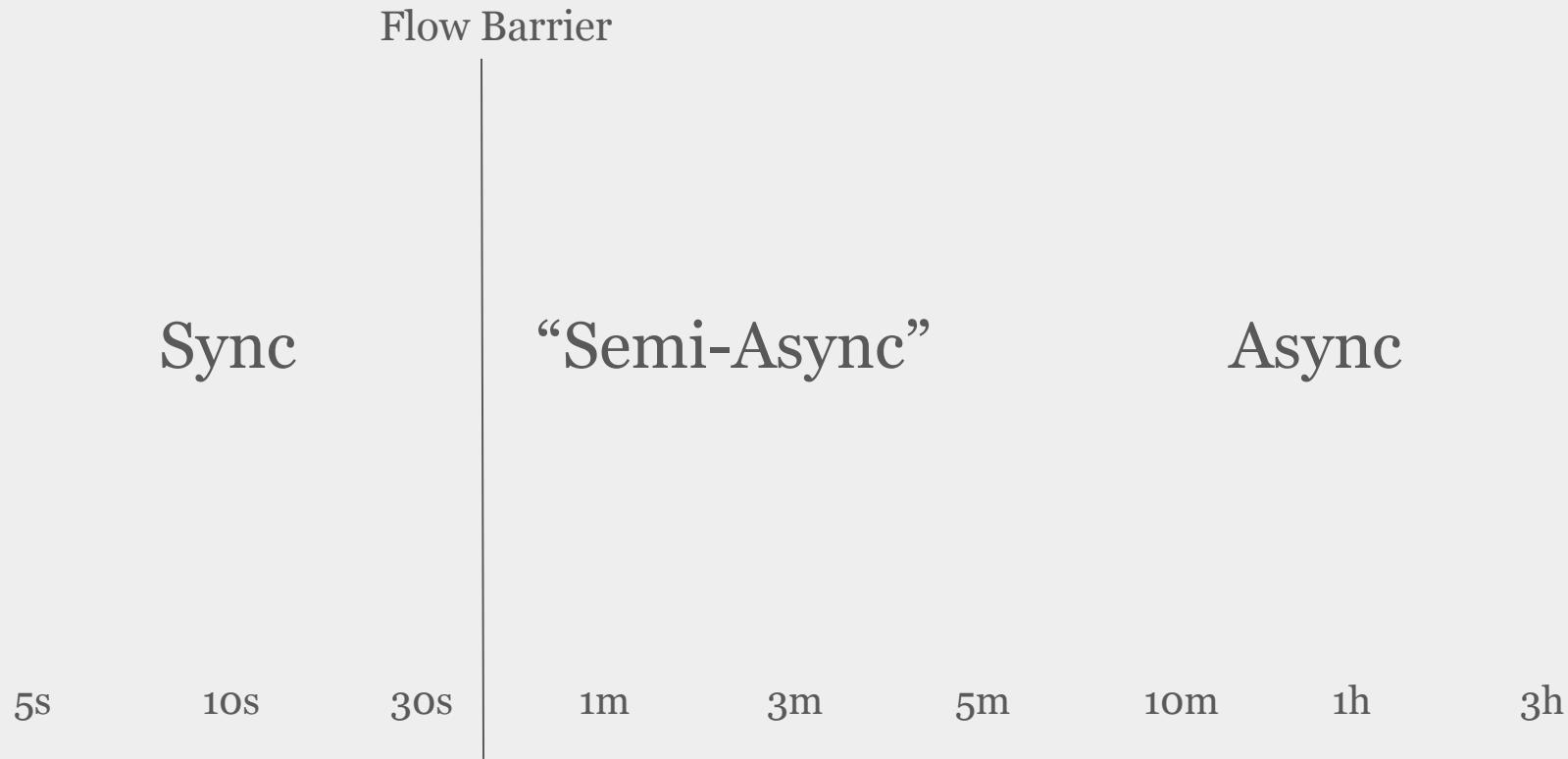
Managing async agents can unlock 10x gains...

...but most people use sync agents.

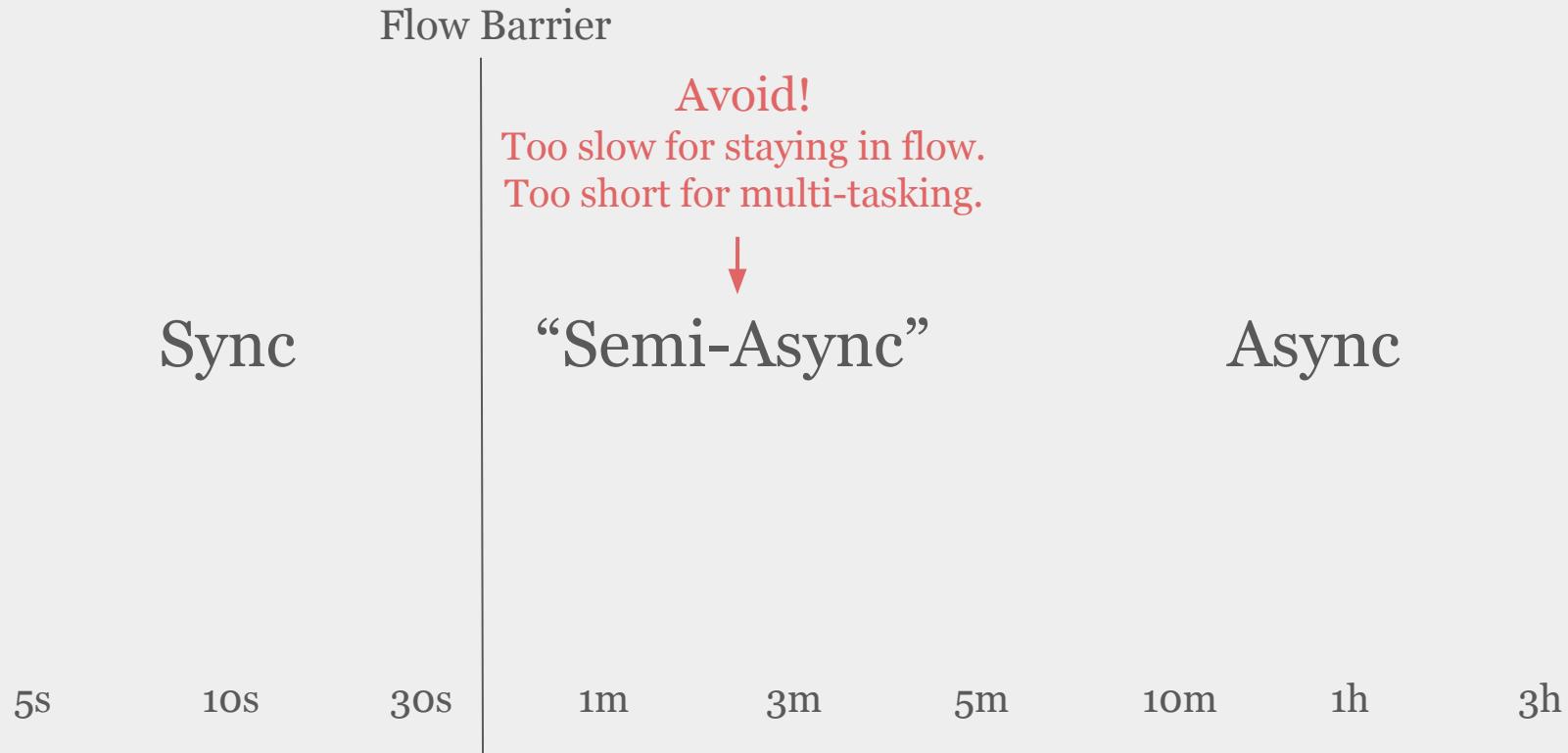
Why?

- Turns out management & delegation is a difficult skill to master – whether it's humans or agents.
- Requires ability to cycle between multiple tasks and quickly understanding new context

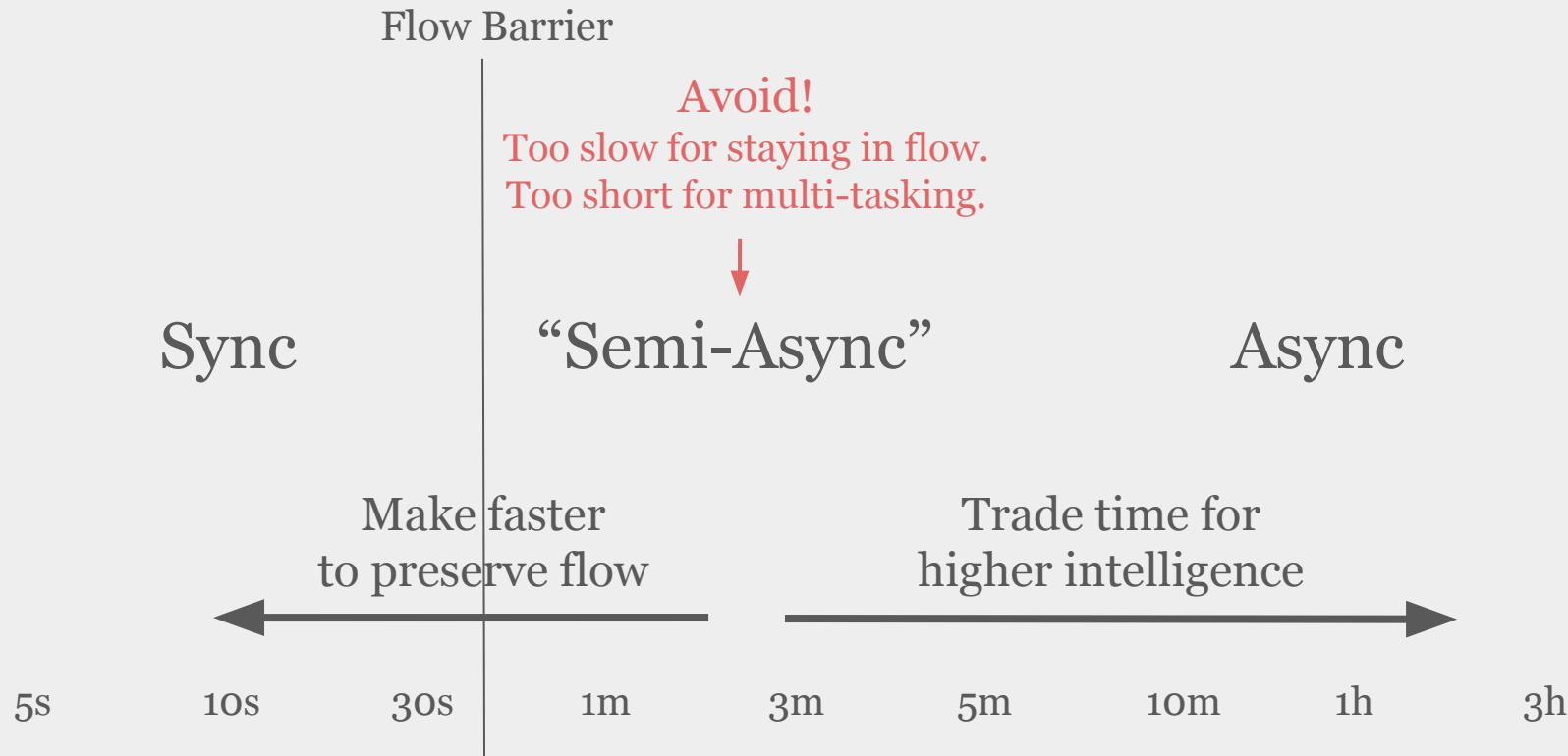
Semi-Async: The awkward middle

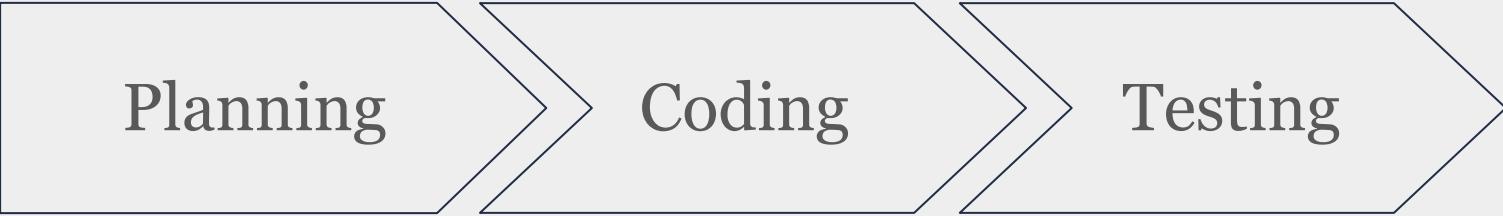


Semi-Async: The awkward middle



Semi-Async: The awkward middle





Planning

Coding

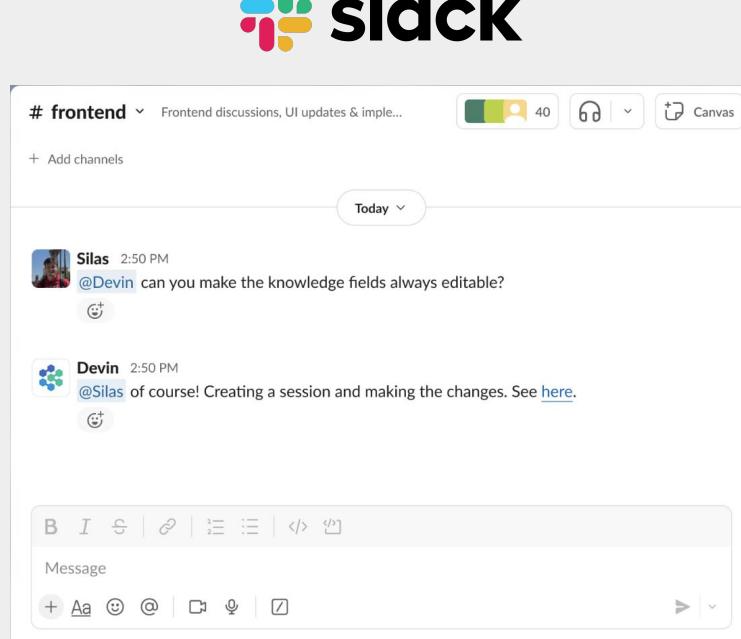
Testing



Planning with Windsurf & Devin

1. DeepWiki
2. Ask Devin
3. Codemaps
4. DeepWiki in Windsurf

Delegate the **coding** to the agent



Delegate the **coding** to the agent



frontend ▾ Frontend discussions, UI updates & imple...

+ Add channels

Today ▾

Silas 2:50 PM @Devin can you make the knowledge fields always editable?

Devin 2:50 PM @Silas of course! Creating a session and making the changes. See [here](#).

Message

+ Aa @ | □ | ↻ | □ | ↻ | □

This screenshot shows a Slack channel named "# frontend". It displays a message from Silas asking Devin to make knowledge fields always editable. Devin responds, stating he will create a session and make the changes, with a link provided. The interface includes standard Slack features like message filtering, a message input field, and a toolbar with various icons.



All Issues ▾

In Progress 10

- ... ○ Implement responsive layout for user dashboard
- ... ○ Fix image rendering bug on mobile product pages
- ... ○ Add modal animation to sign-in/sign-up flow
- ... ○ Style tag filters and improve accessibility in article list
- ... ○ Migrate user sessions table to support UUIDs
- ... ○ Create query to fetch top-rated articles by tag
- ... ○ Refactor post schema to support draft publishing

This screenshot shows a Linear board titled "All Issues". It lists ten tasks under the "In Progress" category. Each task is represented by a small circular icon with a dot, followed by a brief description and a small blue circular icon with a white plus sign. The board has a clean, minimalist design with dark cards and light text.

Testing the agent's changes

Common workflow:

1. Delegate task to Devin (async)
2. Test & refine changes in Windsurf (sync)

Testing the agent's changes

Common workflow:

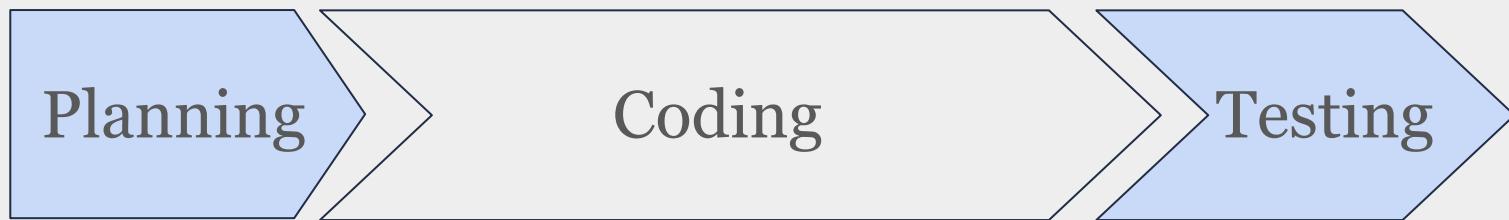
1. Delegate task to Devin (async)
2. Test & refine changes in Windsurf (sync)

Future outlook:

If async agents could test autonomously, the leverage increases.
This is slowly starting to become a reality.

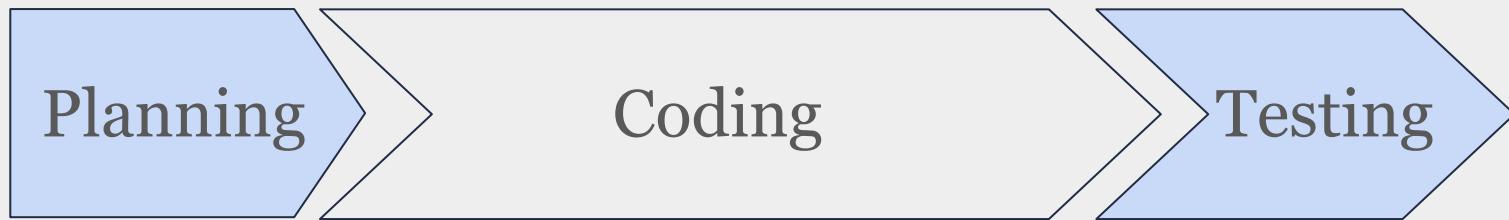
Where are we headed?

Today:

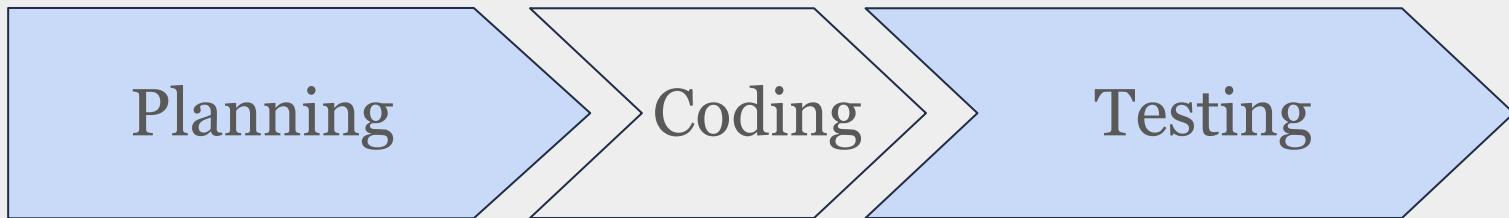


Where are we headed?

Today:



Future:



Where are we headed?

1. The human engineer as the agent manager
 - a. Leveraging sync tools to solve the most difficult problems
 - b. Leveraging async tools to achieve 10x leverage

Where are we headed?

1. The human engineer as the agent manager
 - a. Leveraging sync tools to solve the most difficult problems
 - b. Leveraging async tools to achieve 10x leverage
2. Valuable skills for the future:
 - a. Delegation & multi-threading
 - b. Code reading
 - c. Planning, scoping, architecting

Thank you!