



Lessons Learned From Building GitHub Copilot(s)

Eddie Aftandilian

Principal Researcher, GitHub Next

October 2024



AI devtools are changing software development

- GitHub Copilot launched in June 2021. Today:
 - 1.8M paying users
 - 50% of users' code is written by Copilot
- Next generation of tools are coming
 - Copilot Workspace, Devin, Cursor, et al.



GitHub Next builds AI devtools

- We are a research/advanced R&D team within GitHub
- Created the original GitHub Copilot with OpenAI
- Also created Copilot X and, most recently, Copilot Workspace
- ... and have also tried a bunch of stuff that didn't succeed



The GitHub Next Team



Alice Li
Staff Research Engineer
🔗 [xalili](#)



Amelia Wattenberger
Principal Research Engineer
🔗 [wattenberger](#)



Anthony van der Hooft
Principal Research Engineer
🔗 [avanderhooft](#)



Cole Bemis
Senior Research Engineer
🔗 [colebemis](#)



David Slater
Staff Researcher
🔗 [davidslater](#)



Devon Rifkin
Principal Research Engineer
🔗 [drifkin](#)



Don Syme
Principal Researcher
🔗 [dsyme](#)



Eddie Aftandilian
Principal Researcher
🔗 [eaftan](#)



Eirini Kalliamvakou
Staff Researcher
🔗 [ikalliam](#)



Idan Gazit
Senior Director of Research
🔗 [idan](#)



Jake Donham
Principal Research Engineer
🔗 [jaked](#)



Jonathan Carter
Head of GitHub Next
🔗 [lostintangent](#)



Krzysztof Cieślak
Staff Research Engineer
🔗 [krzysztof-cieslak](#)



Luke Edwards
Staff Research Engineer
🔗 [lukeed](#)



Rahul Pandita
Staff Researcher
🔗 [rahulpandita](#)



Russell Horton
Staff Researcher
🔗 [mrjh](#)



Tamás Szabó
Staff Research Engineer
🔗 [szabta89](#)



Terkel Gjervig
Staff Research Engineer
🔗 [terkeig](#)



Outline

- Organizational lessons
- Technical lessons
- Conclusion



Organizational lessons



ORG1: Think big

GitHub Next

GitHub Next investigates the future of software development.

We are a team of researchers and engineers at GitHub, exploring things beyond the adjacent possible. We prototype tools and technologies that will change our craft.

We identify new approaches to building healthy, productive software engineering teams.



A small team can build a big thing

Introducing GitHub Copilot: your AI pair programmer

Today, we're launching a technical preview of GitHub Copilot, a new AI pair programmer that helps you write better code.

Technical preview

Your AI pair programmer

```
fetch_pic.js  push_to_git.py  JS_d3_scale.js  JS_fetch_stock.js  JS_material_ui.js

1 const fetchNASAPictureOfTheDay = () => {
2   return fetch('https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY', {
3     method: 'GET',
4     headers: {
5       'Content-Type': 'application/json',
6     },
7   })
8   .then(response => response.json())
9   .then(json => {
10    return json;
11  });
12 }
```

GitHub Copilot

Nat Friedman · @nat

June 29, 2021 | Updated February 23, 2022 | 1 minutes

Share: [X](#) [f](#) [in](#)



Non-adjacency

Extension: GitHub Copilot ×



GitHub Copilot v1.234.0

GitHub [github.com](#) | 20,606,218 | ★★★★★ (1077)

Your AI pair programmer

Disable

Uninstall

Switch to Pre-Release Version

Auto Update

DETAILS FEATURES CHANGELOG EXTENSION PACK

GitHub Copilot - Your AI pair programmer

GitHub Copilot is an AI pair programmer tool that helps you write code faster and smarter.

[Sign up for a GitHub Copilot free trial](#)



ORG2: Deliver tangible results frequently

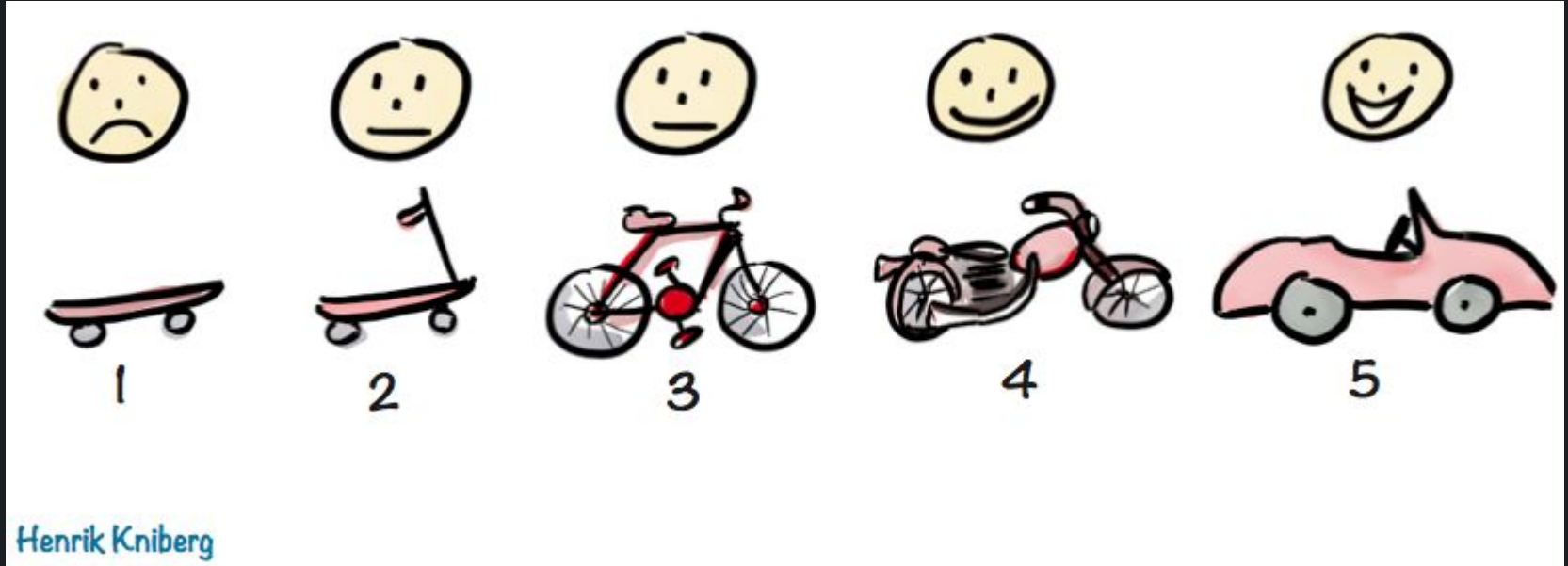
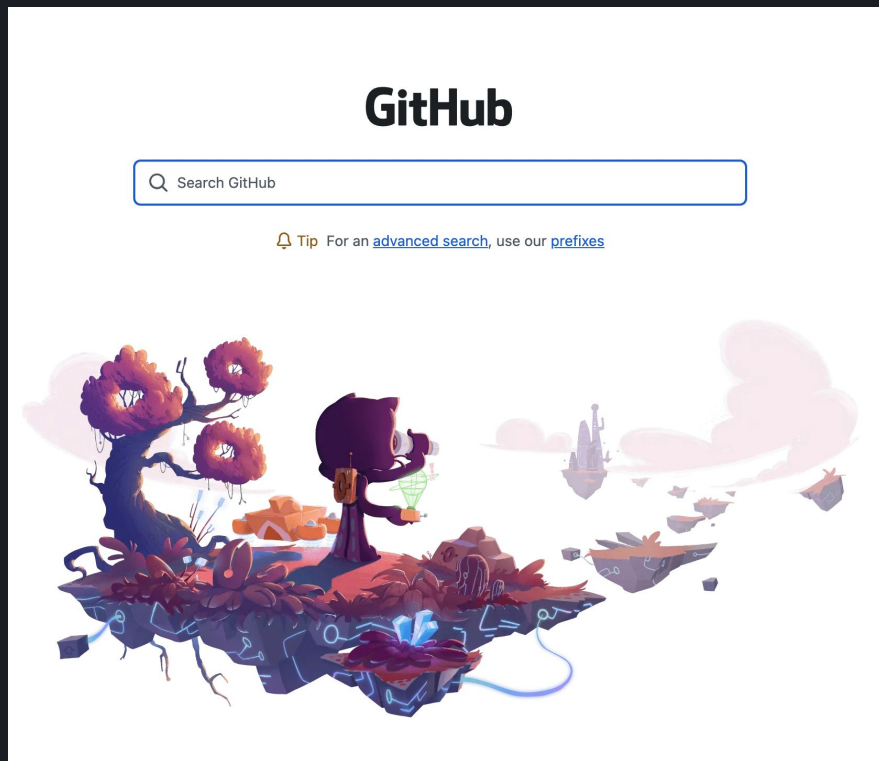


Image credit: <https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>



Retrieval application 1: NL code search



Retrieval application #2: customized Copilot

GitHub Copilot for *Your* Codebase

When editing, GitHub Copilot only knows about the contents of your current file and possibly a few other open tabs, rendering it blind to important type definitions, patterns and greater connections in your codebase. We want to let GitHub Copilot see your entire repo when it comes up with its suggestions.

WHAT'S IT FOR?

Getting better, more relevant
Copilot suggestions

SHARE



STAGE

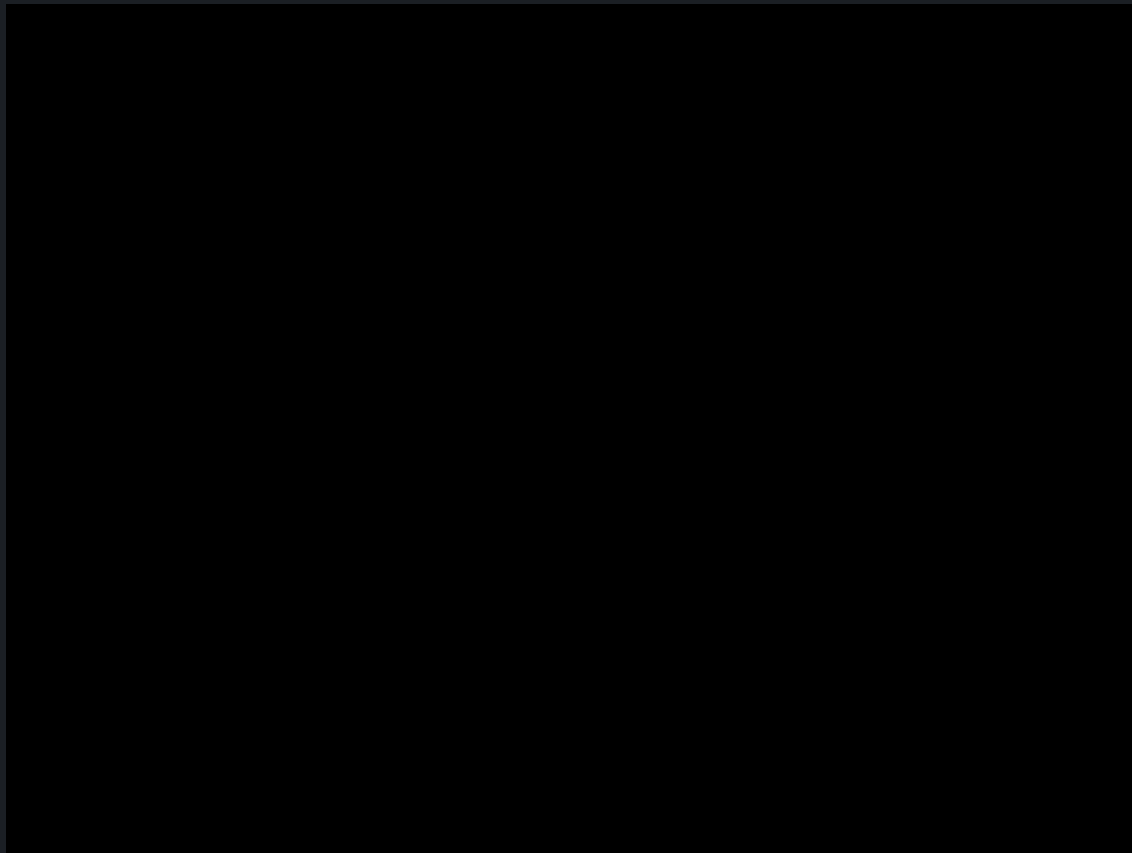
COMPLETED

WHO MADE IT?

-  Eddie Aftandilian
-  Johan Rosenkilde
-  Amelia Wattenberger
-  Vincent Hellendoorn



Retrieval application #3: Copilot for Docs



ORG3: Create urgency via deadlines



ORG4: Be disciplined in project selection and continuation





SpecLang

Can we develop software entirely in natural language, and let an AI-powered toolchain manage the implementation?

WHAT'S IT FOR?

End-to-end development in natural language


SHARE



STAGE


RESEARCH PROTOTYPE

WHO MADE IT?

 Johan Rosenkilde

 Russell Horton

 Krzysztof Cieślak

 Jonathan Carter



ORG5: Beware of long term project ownership



Technical lessons



TECH1: UX is critical in AI developer tools



Copilot Workspace UI prototype

Extract, Edit, Apply

Enter the pivot to extract

Choose a repo:

the behaviour

Extract

1- The code is a command-line application that allows the user to create, read, -list, and remove notes.
2 The notes are stored in a JSON file and have a title and a body. The code uses the yargs module to parse the user's input and the fs module to read and write the file. The code also uses the lodash module for some utility functions. The code defines a separate module called notes.js that contains the logic for manipulating the notes. The code also prints out messages and notes to the console using a logNote function.

1+ The code is a command-line application that allows the user to create, read, -list, count, and remove notes.
2 The notes are stored in a JSON file and have a title and a body. The code uses the yargs module to parse the user's input and the fs module to read and write the file. The code also uses the lodash module for some utility functions. The code defines a separate module called notes.js that contains the logic for manipulating the notes. The code also prints out messages and notes to the console using a logNote function.

Apply

Generated change

```
75 | var noteRemoved = notes.remove(argv.title); //  
    remove a note  
76 | var message = noteRemoved ? "Note removed" : "remove command failed";  
77 | console.log(message);  
78 |  
79 | }  
80 |  
  
81 | else{  
82 |   console.log('command note recognized');  
83 | }  
84 |
```

```
75 | var noteRemoved = notes.remove(argv.title); //  
    remove a note  
76 | var message = noteRemoved ? "Note removed" : "remove command failed";  
77 | console.log(message);  
78 |  
79 | }  
80 |  
81+ | else if (command === 'count') { // added a new branch for counting notes  
82+ |   var count = notes.count(); // call the count function from notes module  
83+ |   console.log(`You have ${count} note(s).`); // print the result  
84+ | }  
85+ |  
86 | else{  
87 |   console.log('command note recognized');  
88 | }  
89 |
```



Copilot Workspace UI v1

Copilot Workspace Experimental

lostintangent/contributor-gallery #6 ... Share

Files changed 2 Split Unified Build succeeded Re-run Create pull request

Topic

Is the contributor's GitHub handle displayed over their avatar, when their gallery cell is active?

Specification

Current

- No, the GitHub handle is not displayed on the active gallery cell
- The code only renders an image element for the contributor's avatar

Proposed

- Yes, the GitHub handle is displayed on the active gallery cell
- The code renders an image element for the contributor's avatar and a text element for the handle value
- The text element is a styled component that:
 - is a sibling of the image element
 - accepts the login value as child content
 - is conditionally displayed when the cell is active
 - has the same font size as the cell size theme property
 - has a black text shadow
 - is centered and has a z-index of 11

Plan

- Define a styled component for the GitHub handle
- Import the cell size theme property in the ContributorCell component
- Add conditional rendering of the GitHub handle in the ContributorCell component
- Update the readme to call out this shiny new feature!

File changes 2

- Change src/components/Gallery/ContributorCell.tsx
- Change README.md

Implement

```
Changed src/components/Gallery/ContributorCell.tsx
1 import React from "react";
2 import styled from "styled-components";
3 import { ThemeProps } from "../theme";
4 import { MatrixCell } from "../../model/contributorsMatrix";
5
6 interface ContributorGalleryCellProps {
7   cell: MatrixCell;
8 }
9
10 export default function ContributorGalleryCell({
11   cell,
12   ...props
13 }: ContributorGalleryCellProps): JSX.Element {
14   const cellContent = cell.contributor ? (
15     <FittedImage
16       src={cell.contributor.avatar_url}
17       {...cell}
18     />
19   ) : null;
20   return <Cell>{cellContent}</Cell>;
21 }
22
23 const borderStyle = ({ theme: { borderStyle } }: ThemeProps) => borderStyle;
24 const cellSize = ({ theme: { cellSize } }: ThemeProps) => cellSize;
25
26 const Cell = styled.div<ThemeProps>`
27   border-bottom: ${borderStyle};
```

```
Changed src/components/Gallery/ContributorCell.tsx
1 import React from "react";
2 import styled from "styled-components";
3 import { ThemeProps } from "../theme";
4 import { MatrixCell } from "../../model/contributorsMatrix";
5
6 interface ContributorCellProps {
7   cell: MatrixCell;
8 }
9
10 export default function ContributorCell({
11   cell,
12   ...props
13 }: ContributorCellProps): JSX.Element {
14   const cellContent = cell.contributor ? (
15     <
16       <FittedImage
17         src={cell.contributor.avatar_url}
18         {...cell}
19       />
20       {cell.isActive && <GitHubHandle>{cell.contributor.login}</GitHubHandle>
21     </>
22   ) : null;
23   return <Cell>{cellContent}</Cell>;
24 }
25
26 const borderStyle = ({ theme: { borderStyle } }: ThemeProps) => borderStyle;
27 const cellSize = ({ theme: { cellSize } }: ThemeProps) => cellSize;
28
29 const Cell = styled.div<ThemeProps>`
30   border-bottom: ${borderStyle};
```

```
Changed README.md
1 # GitHub Contributor Gallery
2
3 ![image](https://github.com/lostintangent/contributor-gallery/assets/116461/b5191c96)
4
5 ## Getting Started
6
7 1. Click the big green 'Code' button, and then select 'Open with Codespaces'
8 1. When connected, open the terminal and run 'yarn start' to launch the web app
9 1. Click the 'https://localhost:3000' URL in the terminal in order to browser the ga
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Status App running



Copilot Workspace UI v2

The screenshot displays the Copilot Workspace UI v2 interface. At the top, it shows the workspace name "lostintangent / gitdoc" and the current branch "master". The interface is divided into several sections:

- Task List (Left):** A list of tasks with checkboxes and status indicators. The tasks are:
 - package.json:** Add a new configuration property `gitdoc.excludeBranches` to allow specifying branches to exclude from auto-commits.
 - README.md:** Update documentation to include information on how to use the new `gitdoc.excludeBranches` setting.
 - src/config.ts:** Add a new property that exposes the `gitdoc.excludeBranches` setting.
 - src/extension.ts:** Update the logic in the `checkEnabled` function to check whether the current branch name is excluded or not. Check for changes to the new `gitdoc.excludeBranches` configuration setting, and re-run `checkEnabled`.
- Files changed (Middle):** A list of files that have been modified, including `package.json`, `README.md`, and `src/config.ts`.
- Code Editor (Right):** The main workspace for editing code. It shows the `src/config.ts` file with the following code:

```
33 },
34 get commitValidationLevel(): CommitValidationLevel {
35     return config().get("commitValidationLevel", "
36 },
37 get commitOnClose() {
38     return config().get("commitOnClose", true);
39 },
40 get enabled() {
41     return config().get(ENABLED_KEY, false);
42 },
43 set enabled(value: boolean) {
44     config().update(ENABLED_KEY, value, vscode.Con
45 },
46+ get excludeBranches(): string[] {
47+     return config().get("excludeBranches", []);
48+ },
49 get filePattern() {
50     return config().get("filePattern", "**/*");
51 },
52 get pullOnOpen() {
53     return config().get("pullOnOpen", true);
54 },
55 get pushMode(): PushMode {
56     return config().get("pushMode", "forcePush");
57 },
58 get timeZone(): string | null {
59     return config().get("timeZone", null);
60 },
61 };
```



Copilot Workspace today

The screenshot displays the Copilot Workspace interface. At the top, it shows the workspace name 'Copilot Workspace' with 'Preview' and 'Give feedback' options. The current project is 'lostintangent / contributor-gallery' with 6 files and a file named 'na.in'. A 'Create pull request' button is visible in the top right.

The main area is divided into two panes. The left pane, titled 'Issue', shows the issue title 'Show GitHub handles when a gallery cell is active'. Below the title, there is a description: 'When a gallery cell is active, we expand it and give it a purple border. However, we should also display the contributor's GitHub handle on top their avatar. For example...'. An image of a gallery cell is shown with a red box highlighting the avatar area. Below the image, there are options to 'Include comments' (checked), 'Additional context' (1 URL, 2 comments), and buttons for 'Brainstorm' and 'Generate plan'. The right pane, titled 'Brainstorm', shows a list of steps to solve the issue:

- **Create a new styled component for the login text:** Define a new styled component that will display the contributor's GitHub handle. This component should be a sibling of the avatar image component and should accept the login value as child content.
- **Update the `ContributorGalleryCell` component:** Modify the `ContributorGalleryCell` component in `src/components/Gallery/ContributorGalleryCell.tsx` to conditionally display the login text when the cell is active.
- **Style the login text:** Set the font size of the login text to the value of the `cellSize` theme property, center the text, and render a black text shadow around it. Also, set the `z-index` to 11 so that it appears on top of the avatar image.

Below the list, it says 'Here is a detailed plan:'

- **Create a new styled component for the login text:**
 - In `src/components/Gallery/ContributorGalleryCell.tsx`, create a new styled component named `LoginText` that will display the contributor's GitHub handle.
- **Update the `ContributorGalleryCell` component:**
 - Import the `LoginText` component.
 - Modify the `ContributorGalleryCell` component to conditionally render the `LoginText` component when the cell is active. Pass the `login` value as child content to the `LoginText` component.
- **Style the login text:**
 - Set the font size of the `LoginText` component to the value of the `cellSize` theme property.
 - Center the text within the cell.
 - Add a black text shadow around the text.
 - Set the `z-index` of the `LoginText` component to 11.

At the bottom, it states: 'By following these steps, you will be able to display the contributor's GitHub handle on top of their avatar when a gallery cell is active. This will enhance the visual representation of the contributors in the gallery.'

At the bottom of the interface, there is a 'Suggested questions' section with a search bar and a list of questions, including 'How do I define a ne... Ask a question...'



TECH2: Build a prototype ASAP



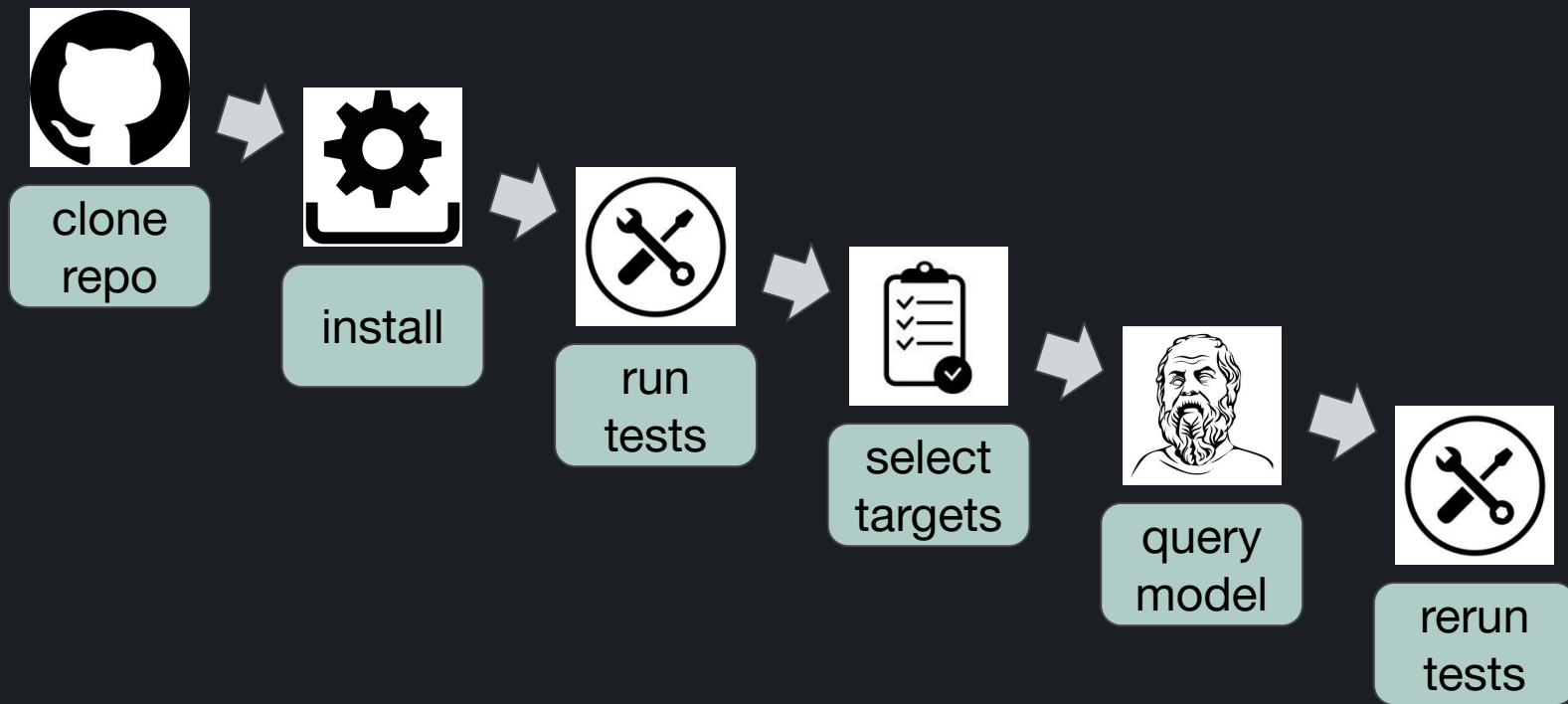
TECH3: Do the simplest thing first



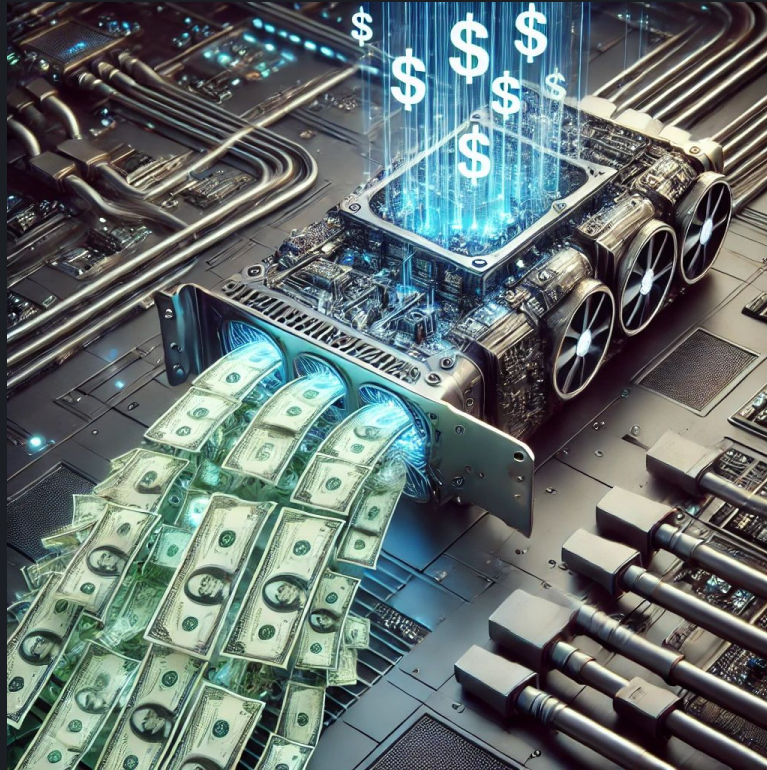
vs.



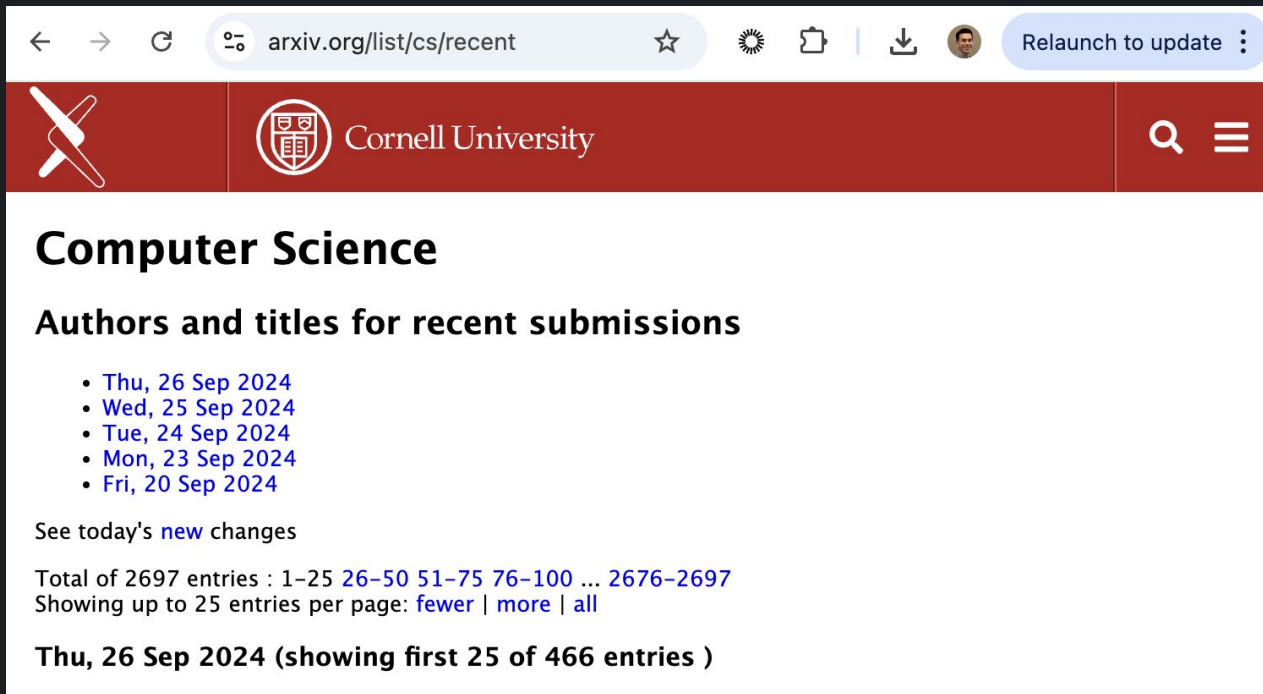
TECH4: Build an evaluation suite



TECH5: Don't over index on cost






TECH6: Keep an eye on the literature



The screenshot shows a web browser displaying the arXiv.org website. The address bar shows the URL `arxiv.org/list/cs/recent`. The page header features the Cornell University logo and name. The main content area is titled "Computer Science" and "Authors and titles for recent submissions". It lists recent submission dates from September 20, 2024, to September 26, 2024. Below the list, it indicates that there are 2697 total entries and shows the first 25 entries for the current date, September 26, 2024.

← → ↻ 🔍 arxiv.org/list/cs/recent ☆ 🌞 📁 | 📄 👤 Relaunch to update ⋮

 Cornell University  

Computer Science

Authors and titles for recent submissions

- [Thu, 26 Sep 2024](#)
- [Wed, 25 Sep 2024](#)
- [Tue, 24 Sep 2024](#)
- [Mon, 23 Sep 2024](#)
- [Fri, 20 Sep 2024](#)

See today's [new](#) changes

Total of 2697 entries : [1-25](#) [26-50](#) [51-75](#) [76-100](#) ... [2676-2697](#)
Showing up to 25 entries per page: [fewer](#) | [more](#) | [all](#)

Thu, 26 Sep 2024 (showing first 25 of 466 entries)



TECH7: Dogfood intensively



Addressing dogfood feedback in Copilot NES

```
def greet(name, args):  
    print(f"Greetings, {name}!")  
    if args.formal:  
        print(f"Good day, {name}.")  
    else:  
        print(f"Greetings, {name}!")
```



```
def greet(name):  
    print(f"Greetings, {name}!")  
  
if __name__ == "__main__":  
    parser = argparse.ArgumentParser("program")  
    parser.add_argument("name")  
    parser.add_argument(["formal"])  
  
def greet(name, formal):  
    if formal:  
        print(f"Hello, {name}!")  
    else:  
        print(f"Hi, {name}!")
```



Conclusion

Think big

GitHub Next

GitHub Next investigates the future of software development.

We are a team of researchers and engineers at GitHub, exploring things beyond the adjacent possible. We prototype tools and technologies that will change our craft. We identify new approaches to building healthy, productive software engineering teams.



UX is critical in AI developer tools



Dogfood intensively

