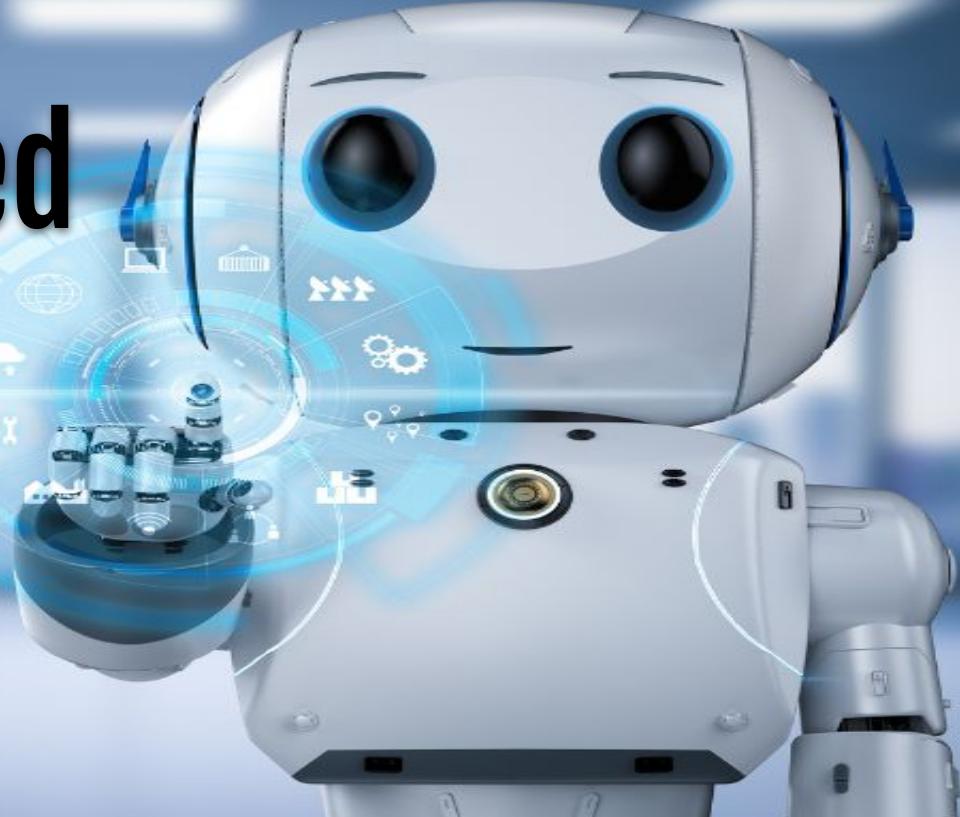


AI for Automated Testing

Kandimalla Hemanth
Knowledge Associate
C-DAC, Bangalore



Outline

01

Evolution of testing and the rising need for AI

00:00-5:00 Min's

02

Core advantages and AI-driven testing features

05:00-10:00 Min's

03

Leading AI tools, spotlight on Cursor AI + live demo

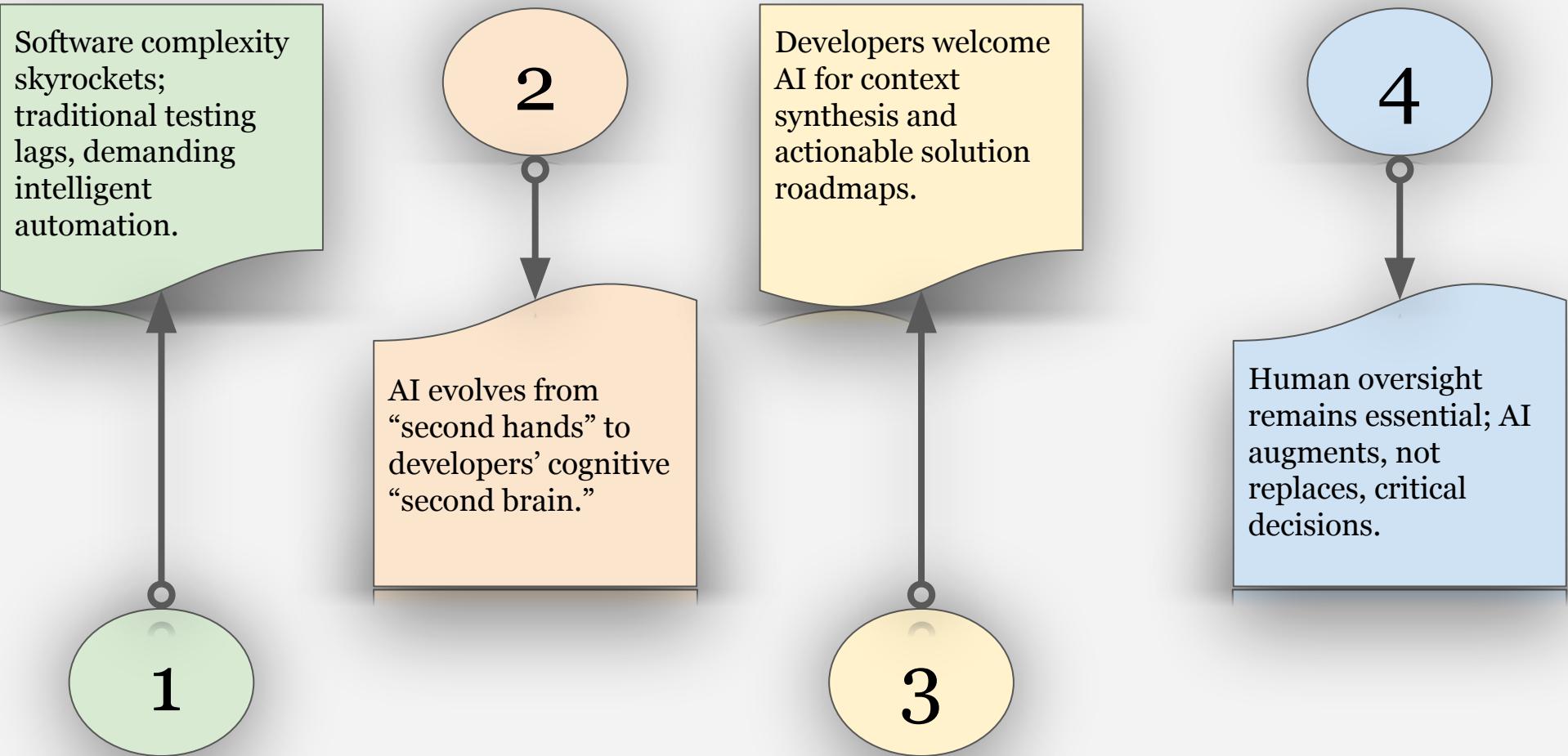
10:00-20:00 Min's

04

2025 insights: impact, adoption paths, next steps

20:00-25:00 Min's

Evolution of testing and the rising need for AI



Software complexity skyrockets; traditional testing lags, demanding intelligent automation.

Big Code's getting bigger.

77% of devs say their codebase grew 5x over the past three years.



Devs are underutilized.

Devs spend only 14% of their time in the codebase writing new code for core products. The rest is spent trying to search, understand, and fix code.



AI is taking software by storm.

95% of devs report using AI tools to write code, but 76% of devs already struggle with code created by someone else that is confusing or difficult to understand.



Software complexity skyrockets; traditional testing lags, demanding intelligent automation.

Innovation velocity is more critical than ever, but 72% say that Big Code presents a real risk to their company's ability to innovate and compete.

The rise of AI-powered dev tools brings in a new set of opportunities and challenges for today's engineering teams, and while GitHub Copilot, ChatGPT, Cody, and other AI coding assistants save devs time, the need to address growing tech debt, code sprawl, and new code creation in the rise of AI has left many teams feeling a mix of excitement and overwhelm.

The Big Code struggle is real:

95%

Need help getting up to speed and stay on top of the codebase significantly faster.

91%

Want to be able to ID and resolve code issues more efficiently.

91%

Would save a significant amount of time if their codebase was fully searchable across all sources and repos.

88%

Want a tool that allows them to have a greater output with fewer people and resources.

Software complexity skyrockets; traditional testing lags, demanding intelligent automation.

Big Code in the AI era.

65% of devs and eng leaders said they didn't have a plan for Big Code prior to the rise of AI. And with AI already broadly adopted across companies—95% of devs say they use GitHub Copilot, ChatGPT, Cody, or similar AI tools to write code—the lack of a plan will have real consequences.

76% of developers are excited about the rise of dev tools powered by AI, but there are still big concerns around AI and its impact on Big Code:



61% are concerned about AI's impact on tech debt.



67% expressed concern about code sprawl due to the growth of AI.

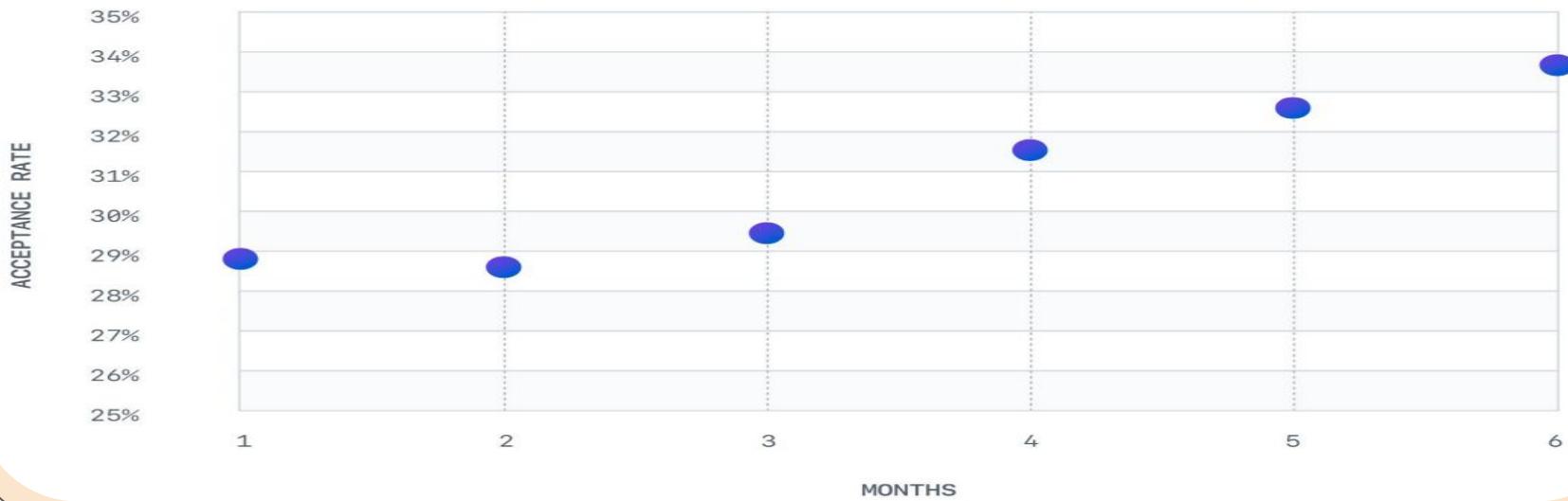


76% are worried about the amount of new code will be created that will then need to be managed.

AI evolves from “second hands” to developers’ cognitive “second brain.”

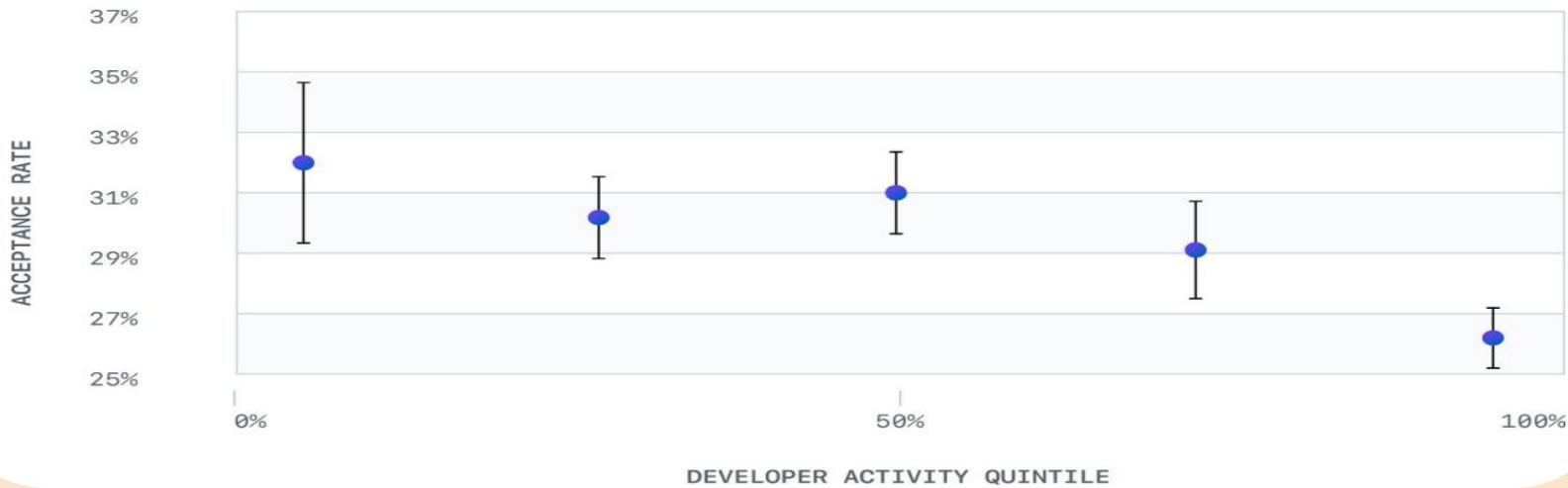
The impact is large and accelerating

Figure 2: Increase in acceptance rate of GitHub Copilot recommendations over time



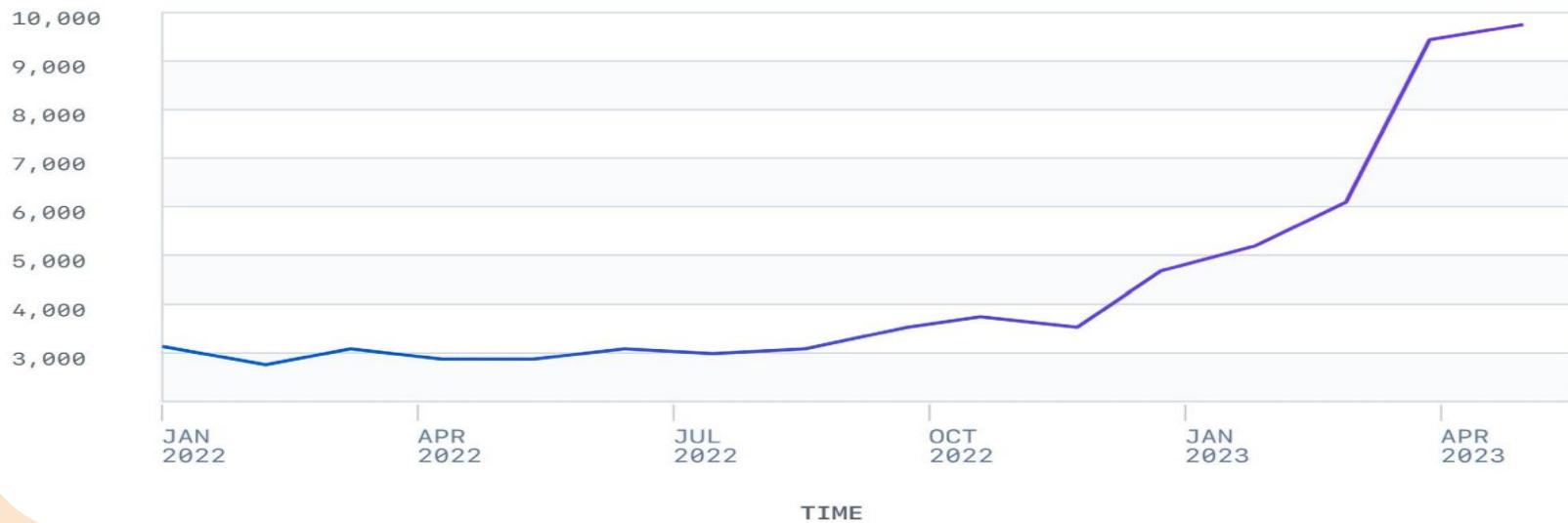
AI evolves from “second hands” to developers’ cognitive “second brain.”

Figure 3: Developers with less experience benefit relatively more than more experienced developers



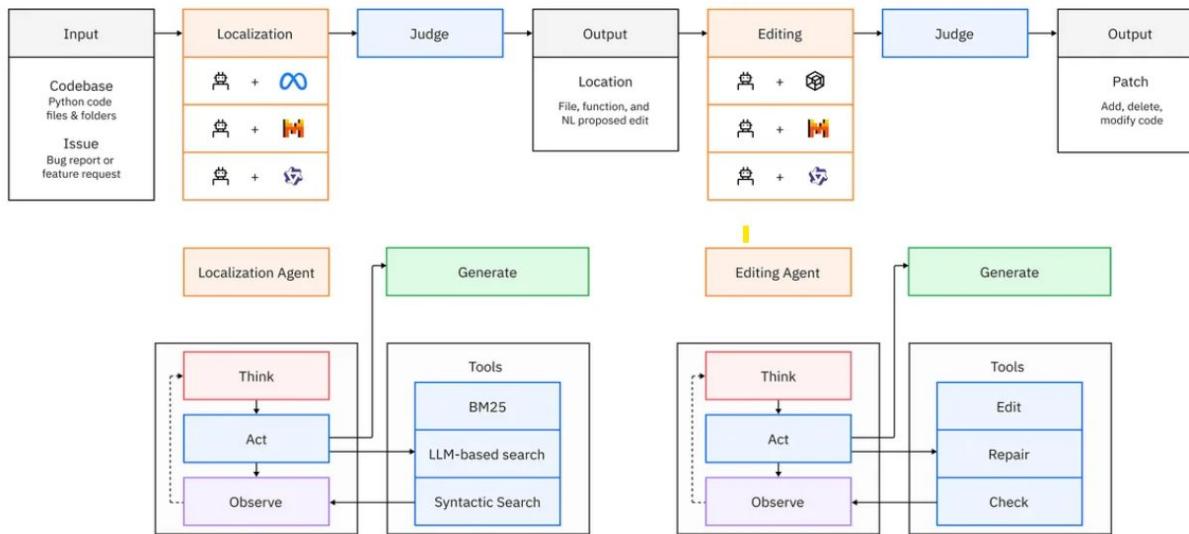
AI evolves from “second hands” to developers’ cognitive “second brain.”

Figure 8: Monthly growth in the number of contributors in generative AI repositories on GitHub



Developers welcome AI for context synthesis and actionable solution roadmaps.

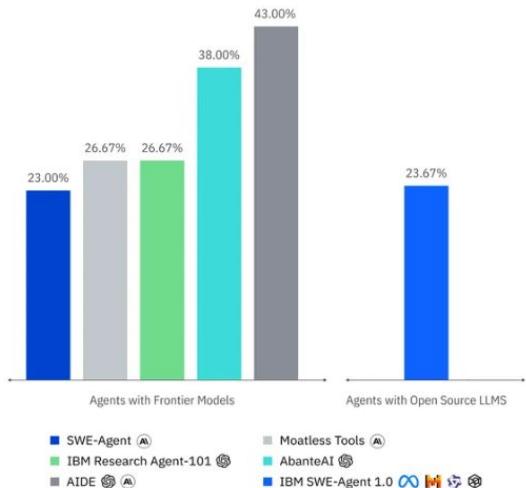
IBM SWE-Agent 1.0



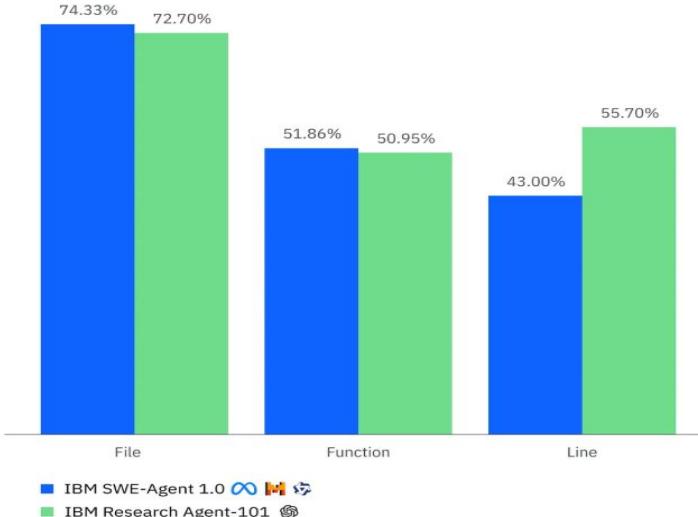
The IBM AI Agent SWE-1.0 system architecture. Given a GitHub issue, the Agent first "localizes" to where the bugs are and then edits those lines of code to resolve them.

Developers welcome AI for context synthesis and actionable solution roadmaps.

SWE-Bench Lite (% resolved issues)



Localization Performance



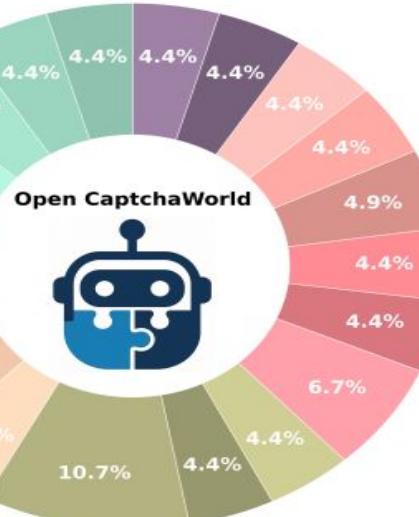
Human oversight remains essential; AI augments, not replaces, critical decisions.

Table 1: Performance of different MLLM backbones within the Browser Use baseline agent on OpenCaptchaWorld. Darker “■” indicates higher success rate@1 and darker “■” indicates higher cost.

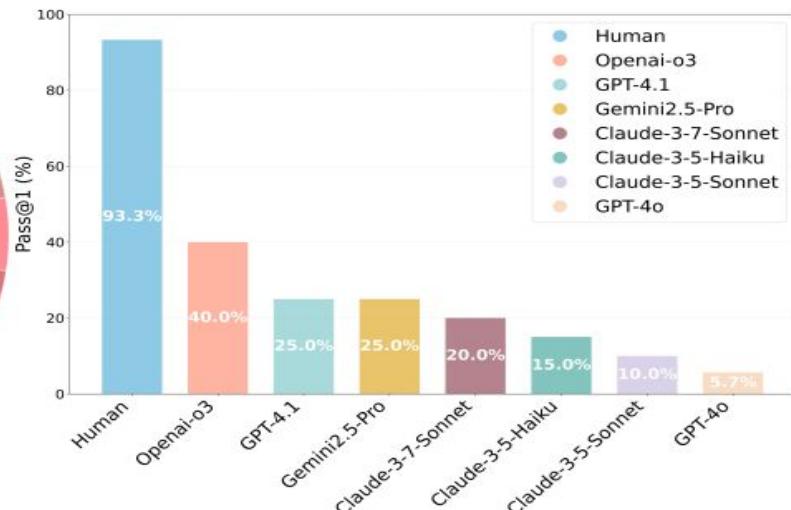
Solver Type	MLLM Backbone	Pass@1 (%)	Cost (\$)
Human	-	93.30	-
	GPT-4o	5.7	25.8
	GPT-4.1	25.0	16.7
	Claude-3.7-Sonnet	20.0	18.7
	Gemini2.5-Pro	25.0	18.1
Browser Use Agents	Openai-o3	40.0	66.4
	Claude-3.5-Haiku	15.0	9.3
	Claude-3.5-Sonnet	10.0	21.9
	Openai-o1	5.0	94.6
	DeepSeek-V3	20.0	7.3

Human oversight remains essential; AI augments, not replaces, critical decisions.

- Select_Animal
- Pick_Area
- Patch_Select
- Object_Match
- Misleading_Click
- Geometry_Click
- Image_Recognition
- Coordinates
- Place_Dot
- Rotation_Match
- Image_Matching
- Connect_icon
- Bingo
- Dart_Count
- Dice_Count
- Slide_Puzzle
- Path_Finder
- Click_Order
- Unusual_Detection
- Hold_Button



(a) CAPTCHAs Distribution



(b) Open CaptchaWorld Leaderboard

Figure 1: **Open CaptchaWorld data distribution and MLLMs performance plot.**

Core advantages and AI-driven testing features

Adaptive Test Resilience:

AI ensures self-healing, evolving test automation.

Strategic Test Optimization:

AI optimizes test focus and UI validation.

Predictive Defect Insights:

AI predicts defects for proactive quality assurance.

Automated Test Creation:

AI automates test case and data generation.

Efficient Operations & Reporting:

AI streamlines reporting, automates routine tasks.

Leading AI tools, spotlight on Cursor AI + live demo

Application development



- Frontend bugs and edge cases
- Unit and E2E testing
- Building SaaS integrations

Personal assistant tasks



- Web research
- Repetitive task automation
- Online booking and reservations

And many others



- Technical debt
- Performance optimization
- Scraping
- New repo onboarding
- Maintaining documentation

Devins are capable of working relentlessly and simultaneously.

Parallel teams of Devins can tackle everything from code refactoring and migrations to frontend bug fixes and Slack requests.

Leading AI tools, spotlight on Cursor AI + live demo

Start with a task ② Delegate to AI ③ Solve problems

The screenshot shows a terminal window titled 'Warp' with a dark theme. It displays three sections of interaction:

- Top Section:** A question from an AI icon: "Can you test the lambda function?". An AI response follows: "Yes, I've created a template.yaml and test json file for you. Please run the code below to test your lambda:". Below this is a command line input field containing the command: "sam local invoke ImageResizeFunction --event event.json".
- Middle Section:** A code editor-like view showing the output of the command. It includes a stack trace: {"errorMessage": "Unable to import module 'lambda_function': No module named 'PIL'", "errorType": "Runtime.ImportModuleError"}
- Bottom Section:** Another AI interaction: "Fix this". The AI responds: "It looks like you don't have the PIL library installed. Please install it in your local directory so it is bundled when you deploy the lambda.". Below this is a command line input field with the command: "Is it ok if I run this command on your machine?". At the bottom right are buttons for '^', 'C' (Cancel), and 'Run'.

Warp is a modern terminal that enhances productivity with AI-powered features like intelligent autocomplete, an IDE-style editor, and customizable themes and keybindings. It streamlines workflows and boosts efficiency for developers.

Leading AI tools, spotlight on Cursor AI + live demo

Build software faster

Intelligent, fast, and familiar, Cursor is the best way to code with AI.

[See more features](#)

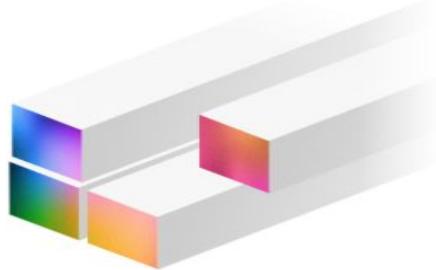
Frontier Intelligence

Powered by a mix of purpose-built and frontier models, Cursor is smart and fast.



Feels Familiar

Import all your extensions, themes, and keybindings in one click.



Privacy Options

If you enable Privacy Mode, your code is never stored remotely. Cursor is SOC 2 certified.



The AI-powered code editor

Designed to boost your productivity to unprecedented levels, Cursor is the ultimate way to write code with AI.

Leading AI tools, spotlight on Cursor AI + live demo

Try Cursor Now



Download for free

Cursor AI



Cursor

is the fastest growing SaaS product.

Reaches \$100 million in Annual Recurring Revenue (ARR), growing 9,900% year-over-year!

Raises \$8 million seed funding from OpenAI's Startup Fund.

Reaches \$65 million ARR

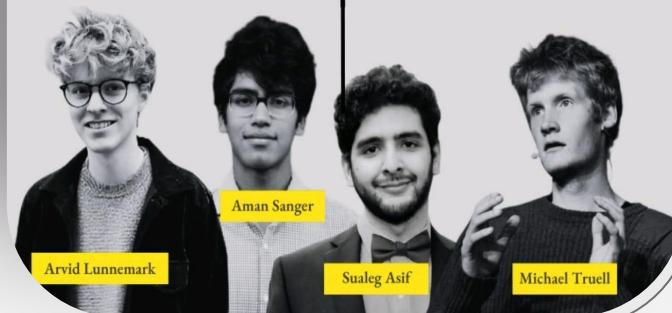
Reaches \$100 million ARR, becomes the fastest growing SaaS company.

JAN 2023
Anysphere, a startup formed by four MIT students launch Cursor.

SEP 2023
AUG 2024
Raises \$60 million Series A from Andreessen Horowitz (a16z), Thrive Capital, among others ; reaches \$19 million ARR

JAN 2025
FEB 2025
Raises \$105 million in Series B, led by a16z, Thrive Capital, and other investors.

Source: Sacra



Agent

Cursor's agent mode can complete tasks end to end. It does this quickly, while keeping programmers in the loop. Try it out by selecting 'agent' in Composer.

CHAT COMPOSER

We should show a strikethrough in the Tab status element when the user has run out of free credit

I'll help you implement the strikethrough styling for the Tab status element when users have run out of free credits. Let me first search for relevant code.

> Searched codebase "Tab status element credits display" + 25 results

Let me search for more specific code related to the Tab status element and credits.

> Searched codebase "Tab status element credits remaining display" + 25 results

Let me search for more specific code related to the credits display.

> Searched codebase "Tab status element credits remaining text" + 25 results

Let me search for one more piece of code related to the credits display.

Now I'll install Tailwind CSS and its dependencies. I'll run the necessary commands to set it up.

```
$ pnpm add _D tailwindcss postcss autoprefixer
```

Skip Run Command ⌘ ⌂

Waiting for approval.

Runs Commands

Cursor can automatically write and run terminal commands. By default, you'll be asked to confirm all commands.

⚠ Property 'testCppConfig' does not exist on type 'I'.
⚠ Property 'testCppConfig' does not exist on type 'I'.

I see that there are some linter errors because the type definitions don't have the testCppConfig field.

> Searched codebase "CppClassResponse type definition or declaration"

Based on the search results I can see that the CppConfigResponse type doesn't have a testCppConfig field. Let me modify the code.

TS ComposerViewsService.ts +12 -4

```
printCppConfig(): void {
  const cppConfig = this.reactiveStorageService.getCppConfig();
  if (cppConfig?.testCppConfig !== undefined) {
    console.log(`Test CPP Config: ${cppConfig?.testCppConfig}`);
  }
}
```

Loops on Errors

Cursor can detect lint errors automatically and apply fixes, reducing the need for manual debugging.

Add tailwindcss to this project

I'll help you add Tailwind CSS to your project. First, let me check if this is a Vite project and examine the current configuration

>Listed directory Current directory 22 results

```
.git
 README.md
 public
 ts
 ts
 no
 pnp
 pa
 RE
```

I see this is a Next.js project. Let's install Tailwind CSS and its dependencies, then configure it. I'll first check the current package.json to ensure we don't have any conflicts.

Read file package.json

Finds Context

Using custom retrieval models, Cursor can understand a codebase. This reduces the need to manually add context.

Tab

Cursor includes a powerful autocomplete that predicts your next edit. Once enabled, it is always on and will suggest edits to your code across multiple lines, taking into account your recent changes.

```
22
23 You, 9 seconds ago - 1 Author (You) 0 implementations
24 pub struct EncodedMessageQueue {
25     pub queue Vec<EncodedMessage ClientFilter>
26
27     sender Arc Sender Vec<EncodedMessage ClientFilter>
28     receiver Arc Receiver Vec<EncodedMessage ClientFilter>
29 }
30
31 impl EncodedMessageQueue for You, 8 seconds ago - Uncommitted changes
32 pub fn new() -> Self {
    let (sender, receiver) = crossbeam_channel::unbounded();
    Self {
        queue: vec![],
        sender: Arc::new(sender),
        receiver: Arc::new(receiver),
    }
}
```

```
// no clue why but this seems to work,
const unbindF5 = () =>
  identifier Perspective INPUT_IDEN
  checkType "code"
}

this.inputs

return () => {
  try {
    unbindKeyC()
    unbindF5() → tab
  } catch (e) {
    // Ignore.
  }
}
```

Cursor Prediction

Cursor predicts your next cursor position so you can seamlessly navigate your code.

```
through(true) BlockBuilder
reduce(true) BlockBuilder
rent_standalone(true) BlockBuilder
|
| name: "Orange Concrete Block").id(5006).bu
| name: "Blue Concrete Block").id(5007).bu
| name: "Red Concrete Block").id(5008).bu
| name: "White Concrete Block").id(5009).bu
| name: "Ivory").id(5012).build(),
| name: "Oak Stairs", id: 5013),
| name: "Ivory Stairs", id: 5013),
```

Multi-Line Edits

Cursor can suggest multiple edits at once, saving you time.

```
const debug = new Debug(document.body,
  dataStyles: {
    top: 10px left 10px fixed
    zIndex: '1000',
    color: '#fff',
    backgroundColor: 'var(--color-primary)'
    padding: '8px',
  },
  dataStyles: {
    top: '10px',
    left: '10px',
    position: 'fixed',
    zIndex: '1000',
  }
}
```

Smart Rewrites

Type carelessly, and Cursor will fix your mistakes.

Chat

Chat lets you talk with an AI that sees your codebase. The chat can always see your current file and cursor, so you can ask it things like: "Is there a bug here?". You can add particular blocks of code to the context with Ctrl+Shift+L or "@." You can chat with your entire codebase with Ctrl+Enter.

CHAT MORE

do you see an issue with `@local_construction.rs` ?

v gpt-4 @ Mention ↗ no context ↗ chat ↗ codebase

Yes, I notice a potential issue in the `delete_file` method. When deleting a file, it only removes the direct file node but doesn't handle nested directories or files that might be children of the deleted node. This could lead to orphaned entries in the `files` BTreeMap.

Here's a fix that removes all child entries from the files map:

local_construction.rs

async fn `delete_file`(
 &mut self,
 file_path: String,
) -> Result<(), anyhow::Error> {
 // ... existing checks ...

 // Remove all child entries from files map

CHAT MORE

physics.rs ↵ reply ▶ apply ⌂ copy

`use std::ops` Apply the entire code

```
use hashbrown::HashMap;
use log::{info, warn, error};
use rapier3d::prelude::CollisionEvent;
use specs::{Entities, ReadExpect, Res}
```

Instant Apply

Apply the code suggestions from chat back into your codebase by clicking the play button on top of any chat codeblock.

How do I configure the physics of a rigid

USING...

↳ Codebase

Reading files...

TS control.ts packages/core/src/core

TS rigid-body.ts packages/physics-engine

JS index.js packages/physics-engine/ex

Codebase Answers

Use @Codebase or Ctrl Enter to ask questions about your codebase. Cursor will search your codebase to find relevant code to your query.

How do I create an @InlineDiffViewZone

v gpt-4 @ Ment

TS inlineDiffViewZone.ts

⌚ renderInlineDiffViewZone

JS promptBarViewZone.css

TS promptBarViewZone.ts

⌚ renderPromptBarViewZone

⌚ Show Commits & PRs

Reference your Code

Reference code with @ symbols to be used as context for the AI. Just type @ to see a list of all the files and code symbols in your folder.

How do I implement a similar UI for this?

✓ gpt-4 @ Mention 

 Screenshot 2024-10-10 at 2.38.23 AM

 Screenshot 2024-10-10 at 2.37.30 AM 

 Screen Recording 2024-10-10 at 2.35.37 

 Screen Recording 2024-10-10 at 2.35.27

Use Images

Hit the image button under chat or drag an image into the input box to include visual context into chat.

@Web What's new in the latest NextJS

USED

 index.tsx Current file

WEB PAGES

 Next.js 15 RC  Releases - vercel/next.js

 Next.js 14 on Vercel - Vercel  Next.js by Vercel

 Release v12.1.7-canary.8 - vercel/next.js

Ask the Web

Get up-to-date information from the internet with @Web. Cursor will search the web for you and use the latest information to answer your question.

Add new doc

ENTRYPOINT

 https://threejs.org/docs/

PREFIX

 https://threejs.org/docs/

NAME

 Three.js

Use Documentation

Reference popular libraries using @LibraryName, or add your own using @Docs → Add new doc.

Ctrl K

Ctrl K lets you edit and write code with the AI. To edit, try selecting some code, click "Edit," and describe how the code should be changed. To generate completely new code, just type Ctrl K without selecting anything.

add an inside case

⌘ ↵ Submit Edit

↖ ↵ quick question ⌘ ↵ edit codebase

⇧ ⌘ K to toggle

```
let buffer = "";
let state: "outside" | "newline" = "outside";
let state: "outside" | "newline" | "inside" = "outside";
for await (const datum of stream) {
  const newToken = datum.content;
  for (const char of newToken) {
    if (state === "inside") {
      if (char === "") {
        state = "outside";
        buffer = "";
      } else {
        buffer += char;
      }
    }
    buffer += char;
  }
}
```

OUTPUT TERMINAL zsh + × ┌ ┘

```
→ portal-website git:(ian-features)
On branch ian-features
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)

      reset the last commit
```

↪ Submit Edit ↩ quick question

do you know why it's -100?

↪ Generate ↩ quick question

↙ gpt-4 ⌂ K to ↴

```
def get_ending_tensor (orig_tensor: torch.Tensor):
    assert orig_tensor.shape[:2] = sequence_ids.
    # print(f'orig tensor - {orig_tensor.shape}')
    # print(f'sequence ids - {sequence_ids.}
```

Terminal Ctrl K

Use Ctrl K in the terminal to write terminal commands in plain english. Cursor will convert them into the terminal command you need.

Quick Questions

If you have any quick questions about certain parts of your code, you can select and click on "quick question" to get your answer right away.

“

More than 70% of our engineers now use it, and they're doing far more than just using autocomplete; over 45% of our code changes are now fully written by AI.

JAMES REGGIO



“

We've seen our developers use Cursor as they would a teammate: instead of simply delegating tasks to coding agents, they are pairing with them through iterations and feedback.

ALEXIS LÊ-QUÔC



“

GitHub Copilot barely reached 20% adoption with our team, but with Cursor we hit nearly 100% usage right after rollout. The before-and-after has been incredible to see: we're shipping about 50% more code.

ANTON ANDREEV



“

Cursor has transformed the way our engineering teams write and ship code, with adoption growing from 150 to over 500 engineers (~60% of our org!) in just a few weeks. Cursor has quickly become an indispensable part of our development toolkit!

ALBERT STRASHEIM



Cursor Demo