

AI-Assisted Software Engineering

Nathan Cassee --- 16 September 2026

Dr. Nathan Cassee

- Postdoctoral researcher at the Department of Computer Science
- Obtained my PhD at Eindhoven University of Technology in the Netherlands

**Emotions &
Decision Making**

AI Adoption

Research

**Mining Software
Repositories**

**Software
Architecture**



The State of AI-Assisted Software Engineering

A brief History of Generative AI in Software Engineering



Github
Copilot

June
2021

Github Copilot (2021)

```
1  from __future__ import print_function
2  import argparse
3  import torch
4  import torch.nn as nn
5  import torch.optim as optim
6  import numpy as np
7  import matplotlib
8  matplotlib.use('Agg')
9  import matplotlib.pyplot as plt
10
11 class Sequence(nn.Module):
    def __init__(self):
        super(Sequence, self).__init__()
        self.lstm1 = nn.LSTMCell(1, 51)
        self.lstm2 = nn.LSTMCell(51, 51)
        self.linear = nn.Linear(51, 1)

    def forward(self, input, future = 0):
        outputs = []
        h_t = torch.zeros(input.size(0), 51, dtype=torch.double)
```

A brief History of Generative AI in Software Engineering



Github
Copilot



ChatGPT



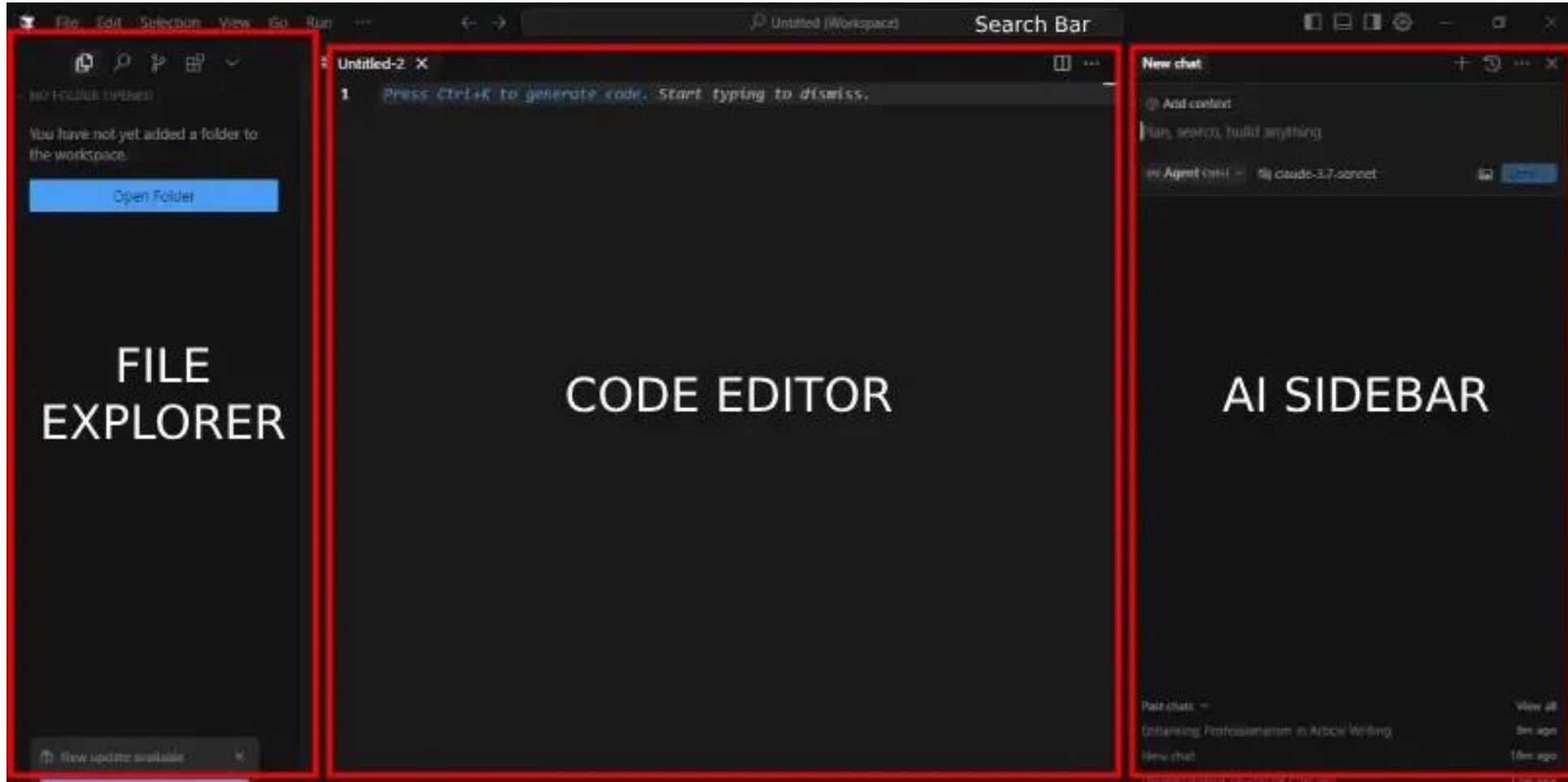
Cursor
IDE

June
2021

November
2022

March
2023

Cursor IDE (2023)



A brief History of Generative AI in Software Engineering

Each new development on this timeline is a new interface, or wrapper, around the same technology. They are being used for different tasks or have enabled new ways of working.



Github
Copilot



ChatGPT



Cursor
IDE



Claude
Code



Xirp

June
2021

November
2022

March
2023

February
2025

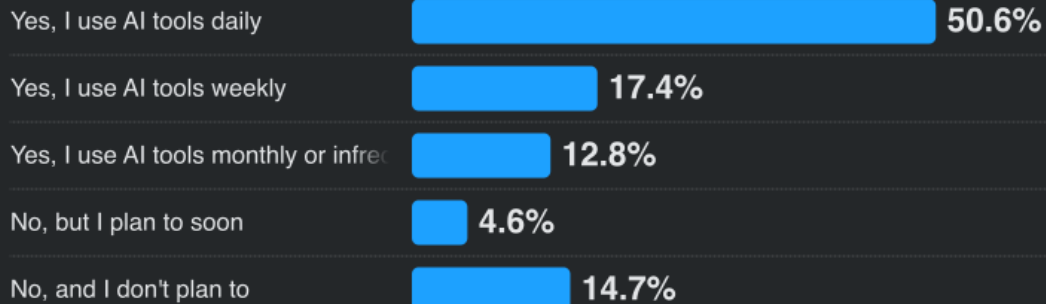
August
2026

AI Adoption by Software Developers

In 2025, 80% of developers in industry reported using AI Tools.

Sentiment and usage / Professional Developers

AI tools in the development process





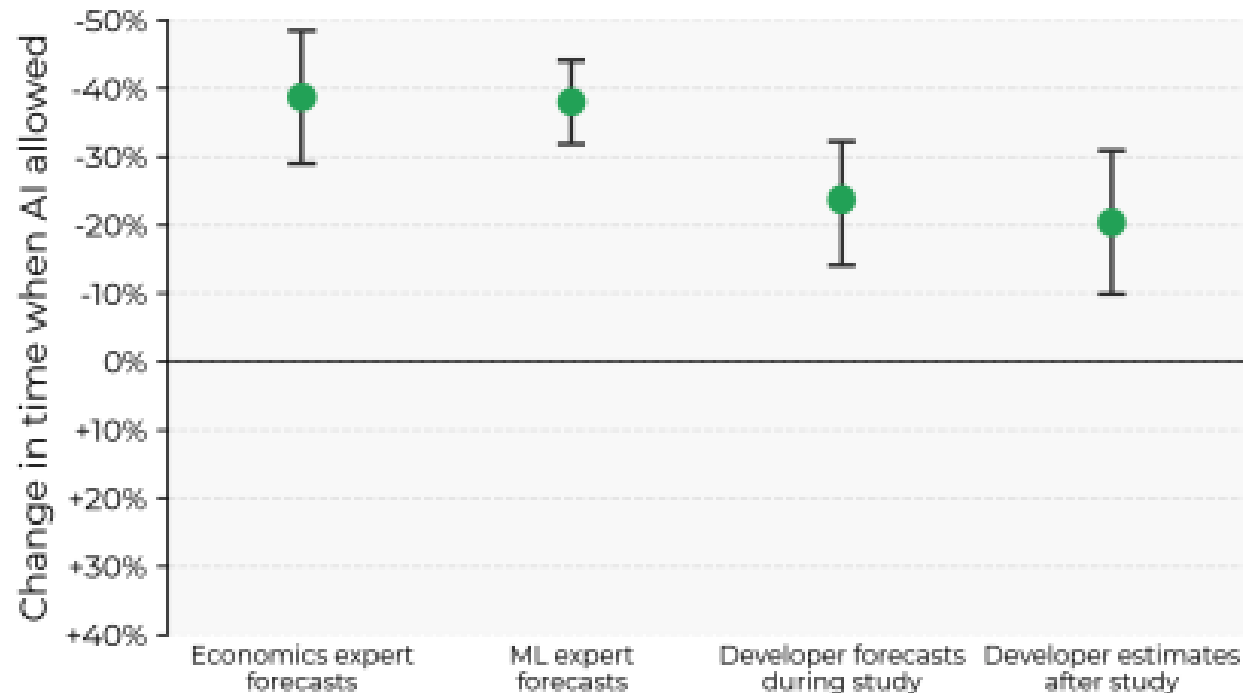
What's the impact of these tools on software development and software developers?



Cursor
IDE

Is AI Making Software Engineers Faster?

In the first half of 2025 METR ran an experiment to measure whether AI tools sped-up software developers.

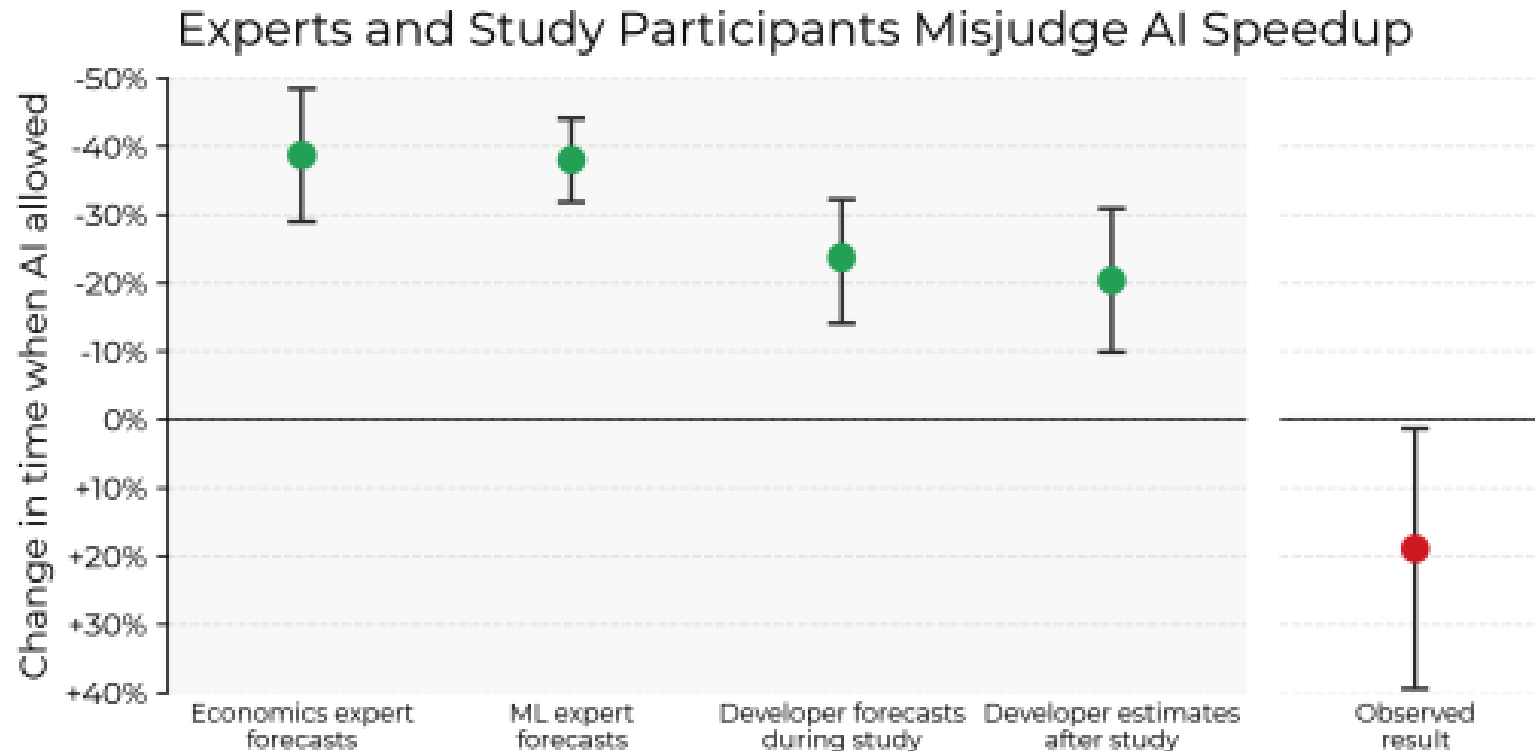




Cursor
IDE

Is AI Making Software Engineers Faster?

In the first half of 2025 METR ran an experiment to measure whether AI tools sped-up software developers.

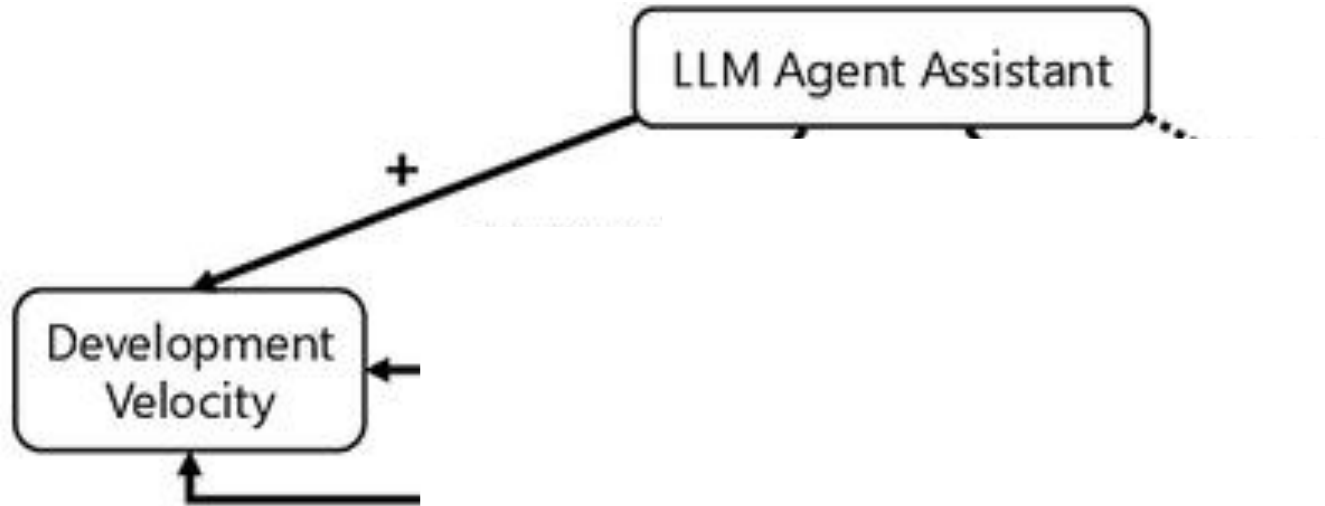




Cursor
IDE

Does AI improve the quality of Codebases?

In 2026 Vasilescu studied how the quality and size of codebases changed when developers used Cursor.

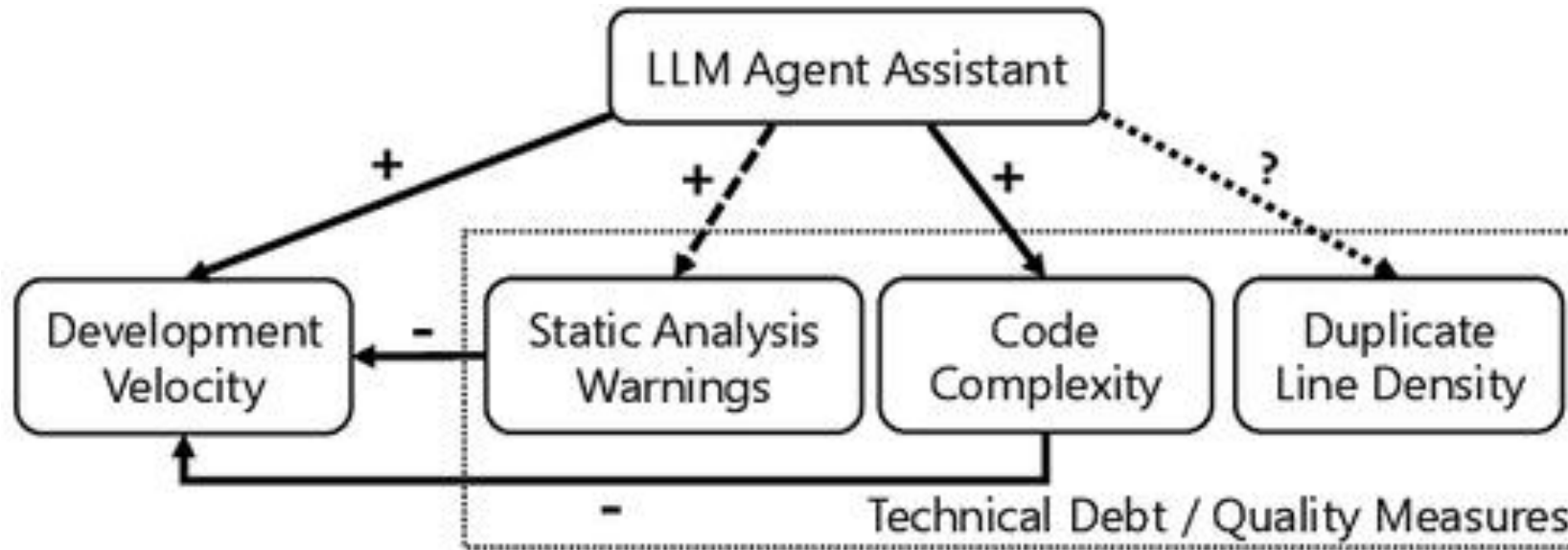




Cursor
IDE

Does AI improve the quality of Codebases?

In 2026 Vasilescu studied how the quality and size of codebases changed when developers used Cursor.



What do developers themselves say about AI?

- In 2025 Miller interviewed 54 developers at Microsoft.
- Frequent Users of AI said working with AI felt like working with a coworker (Collaboration), whereas infrequent users described it was like working with any other tool (Feature).
- Frequent users kept going when AI made mistakes, or experimented with AI, whereas infrequent users were quicker to give up, or more conservative when it came to experimentation.
- These findings show how even at Microsoft there's a lot of different ways people think about, or work with AI.

These three studies show us there's still a lot we don't know about how AI is changing development

- **Is it making developers faster?** Or do developers just think they are more productive?
- **Does it write perfect code?** Or is AI generated by coding agents making code more difficult to maintain?
- How do developers work with AI? How **should** they work with AI? How do we help a new generation of software developers craft maintainable code?

Research Setting: AI-Assisted Software Architecture for 3rd year undergraduates

- **Setting:** Software Architecture course in the Fall Semester of 2025
- **Premise:** Now that companies are expecting “AI-native developers”, engineering education needs to adapt.
- **Assignment:** Design and Implement a room-booking website using modern web technologies.
 - Typescript Front-end, Typescript Backend, Database, all Dockerized.
- **AI Mandate:** Experiment with AI, use it to sharpen decision making, implement boilerplate code, verify and review the architecture.

Research Question: How did students experience working with AI?

- A large study conducted across three courses in 2025 to understand what working with AI was like for the students working on the project.
- For the architecture project, 32 students from 8 teams participated in **focus groups** after completion of the project.
- Examples of focus group questions:
 - *What was it like to work with AI?*
 - *What did you use AI for?*
 - *How did your team make architectural decisions?*

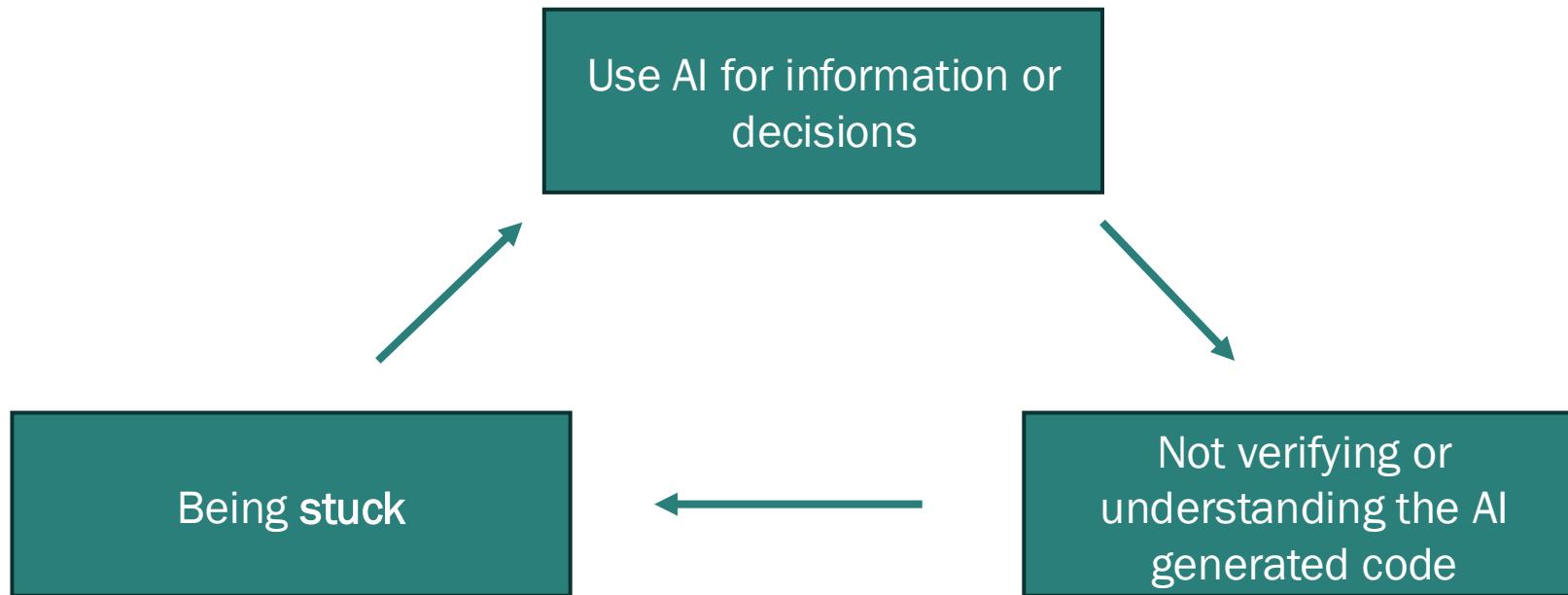


What students told us about surrendering to AI

A short description of what this section covers

AI (over)-reliance

- Some of the participants said that during the project they were trapped in a deepening cycle of AI reliance.



How AI reliance made participants stuck

- Being stuck means being unable to move forward or make progress.
- In 2025 Anderson identified different ways in which AI made developers stuck: By returning incorrect information or suggesting flawed plans. The participants in this study experienced similar things.
- Participant FG50: *“There is a problem I was having where I was trying to make a request to the database and it would fail because of the time was in the wrong format, and I could not figure out why it was in the wrong format”*

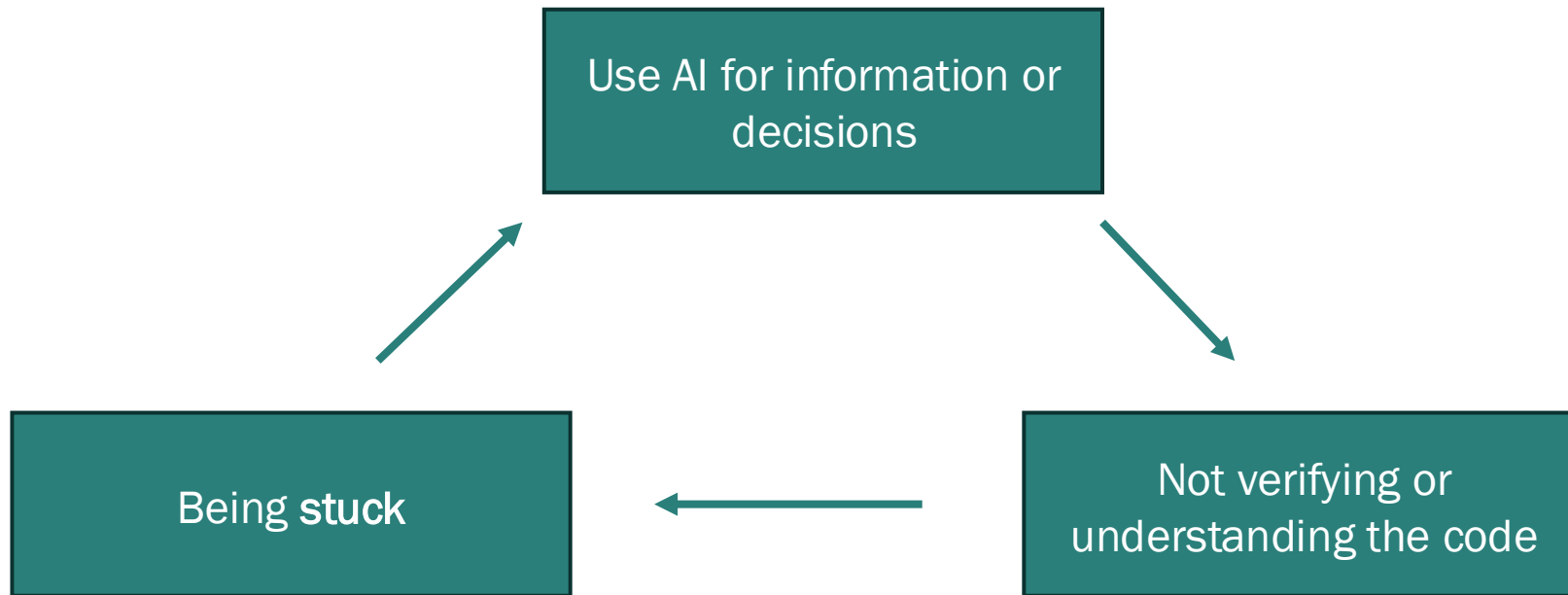
How did the AI reliance make participants feel?

- Shame (P19): *“From my perspective, shame I have to rely on it [AI] rather than my own knowledge.”*
- Being gaslit (FG51): *“The term I started with [to describe AI] was like intelligent gaslighter because honestly it does feel like it then knows a couple of things it does give you it could be very helpful it gives you the answer you sort of want but it just sort of says like I’m right you’re wrong trust me on this it should work the way I tell you to.”*
- Frustrating (P25): *“It [AI] was useful but also frustrating as it would often spew out code that did not work correctly.”*

AI reliance eventually made participants lose motivation

- Most dishearteningly, participants who described being stuck, also said working like this made the project feel empty, or hollow.
- P5: *“But prompting for four months is, like, pointless.”*
- P16: *“That’s a really good point. I feel like I just didn’t care. It just was, like, very emotionally disconnected from the work. Like, I was, like, had no pride in it. I was just, like, just...”*
- P22: *“Yeah, I don’t really take much ownership over it either. Like, it doesn’t feel like my work.”*

AI (over)-reliance

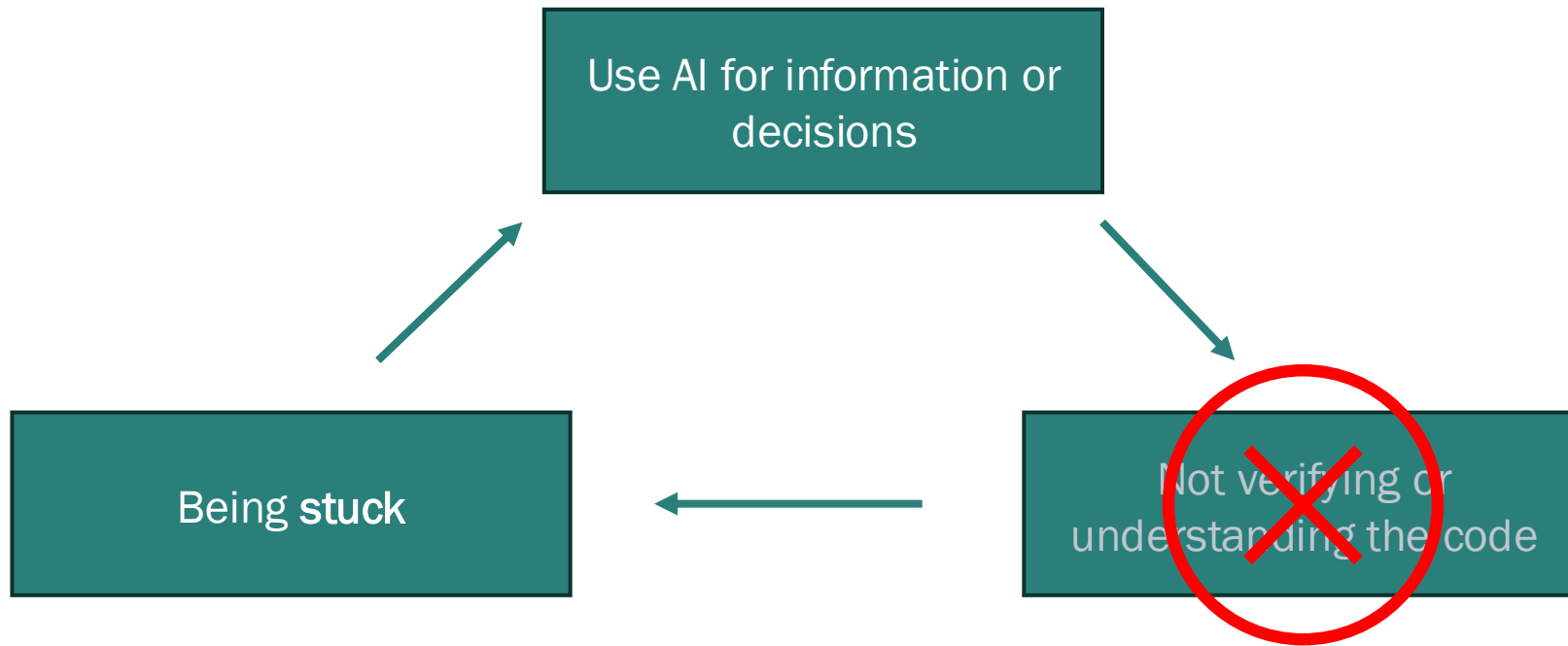




Applying judgment prevented surrender to AI

Judgment helped students break the cycle of AI reliance

- A big reason why participants got stuck was because they did not apply judgment to verify what AI was doing.



Using judgment to break the cycle of AI reliance

- Judgment: Relying on knowledge or experience to evaluate and assess information.
- P44: *“I kept prompting it[AI] to do something again and again. I would usually look through what they had given me and see if I can find any problems like what are the specific errors that I'm encountering or what do I not like about the first way that they [AI] did it and then I would probably like rewrite the prompt to include what I didn't like or liked about the first one or what I thought I had missed context wise.”*

Judgment requires knowledge!

- P3: *“Even in like previous classes before taking this class I’ve seen stuff like that [software diagrams and typescript]”*
- P24: “uh for team [redacted] for implementation II we all spent three hours out of our day and watched a video on nestjs”
- There are different ways to acquire the knowledge needed to apply judgment, but all accounts of students who successfully worked with AI started with knowledge.

With judgment also came taste on how the system should be built

- This went beyond just telling AI what to do.
- P43: *“I don't know if I'd call it a challenge but part of the procedure for having GenAI create good code requires several iterations and revisions. Everything I had it do for me were tasks I knew how to do myself, so it just accelerated my working speed. When it makes mistakes, I catch them and tell it to fix them.”*
- P37: *“Sometimes without doing that [specifically adding context], it[AI] would start trying to make changes in different places that you didn't really want it to.”*

Strategies to apply judgment, shared by participants

1. Ensure AI has the right context and reset or modify the context frequently.

Strategies to apply judgment, shared by participants

1. Ensure AI has the right context and reset or modify the context frequently.

2. Let AI create a plan first, then evaluate and edit the plan.

Strategies to apply judgment, shared by participants

1. Ensure AI has the right context and reset or modify the context frequently.

2. Let AI create a plan first, then evaluate and edit the plan.

3. Let AI work in increments you can oversee and evaluate.

Strategies to apply judgment, shared by participants

1. Ensure AI has the right context and reset or modify the context frequently.

2. Let AI create a plan first, then evaluate and edit the plan.

3. Let AI work in increments you can oversee and evaluate.

4. Use custom instructions to encode your domain.



In Summary

A brief History of Generative AI in Software Engineering

There's a wealth of very powerful AI tools out there to help developers build and implement software systems, but everyone is working hard to figure out how to work with these tools.



Github
Copilot



ChatGPT



Cursor
IDE



Claude
Code



Xirp

June
2021

November
2022

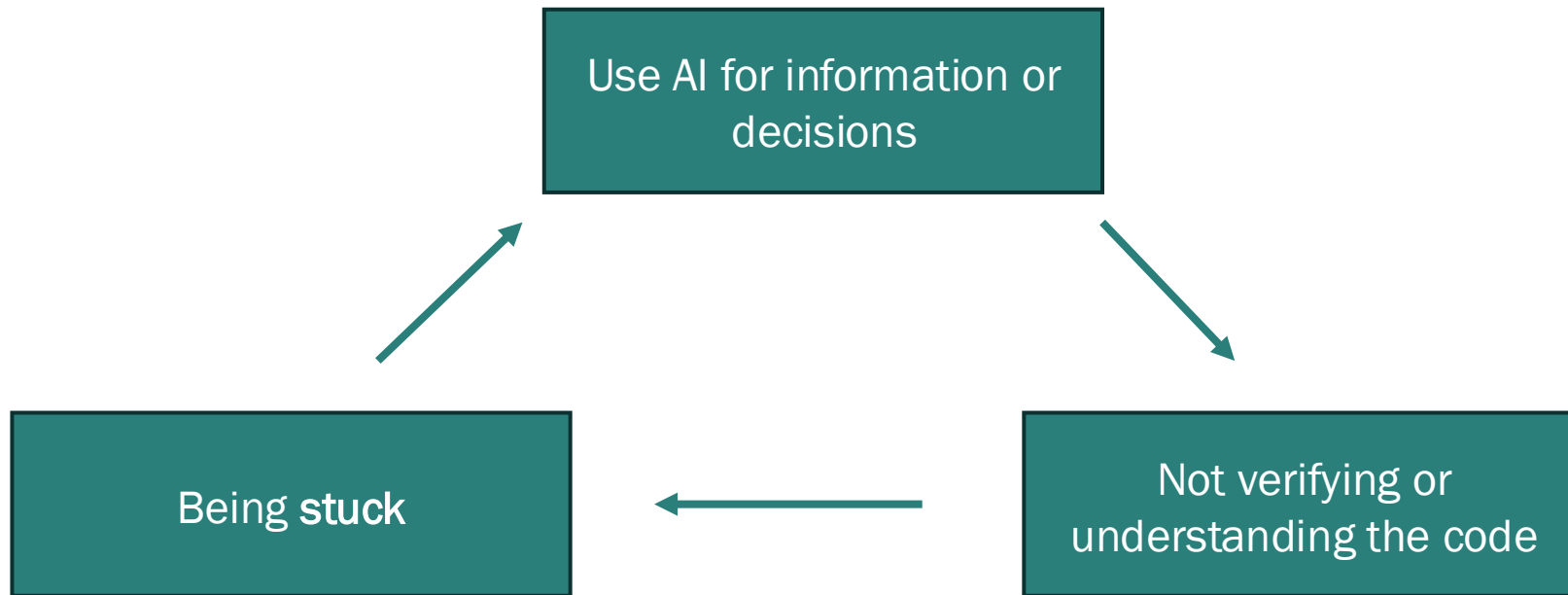
March
2023

February
2025

August
2026

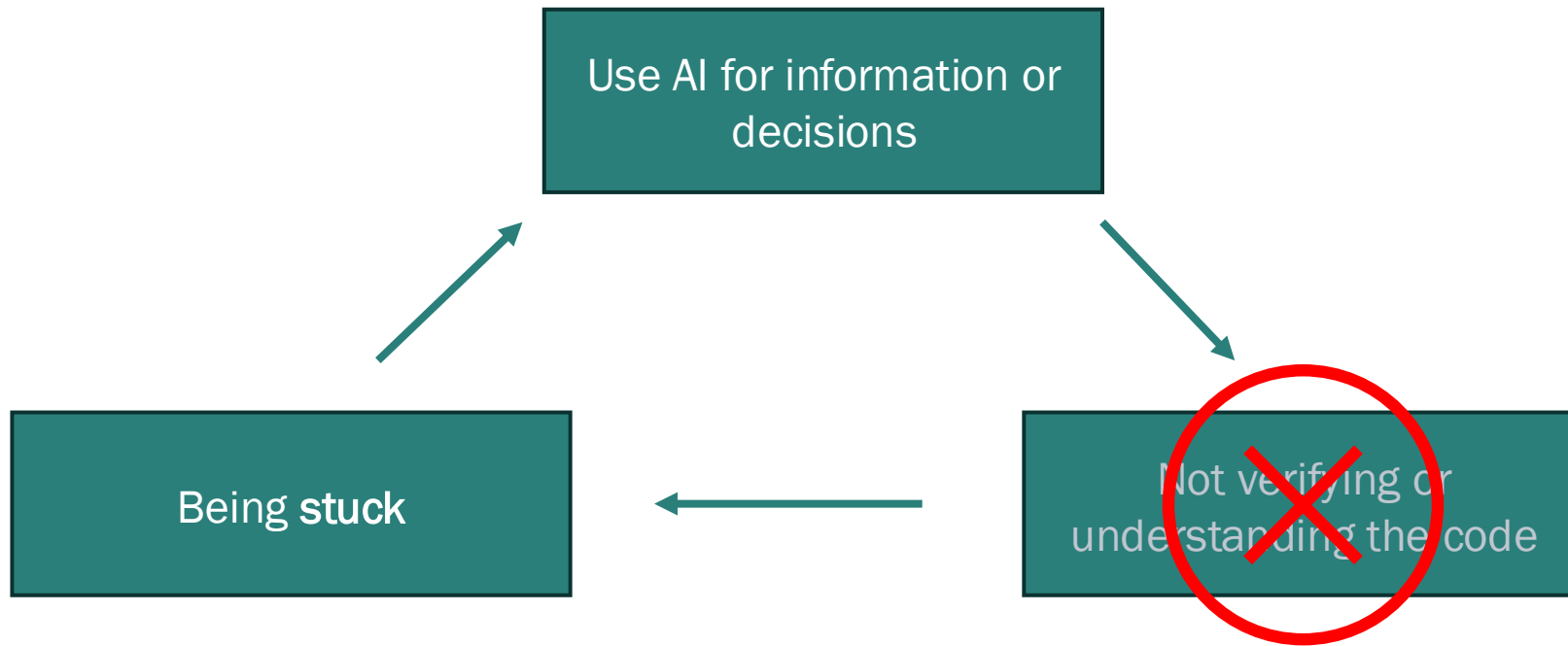
AI (over)-reliance

- It's very easy to get stuck in a cycle of AI reliance, that makes developers feel terrible, and that erodes motivation.



Judgment helped students break the cycle of AI reliance

- This cycle can be broken, but it requires knowledge and experience. Two things built through courses like this!



Thank you

References

- 2025 Stackoverflow Survey: <https://survey.stackoverflow.co/2025/>
- METR Productivity study: <https://arxiv.org/pdf/2507.09089>
- Vasilescu 2026 study:
<https://dl.acm.org/doi/full/10.1145/3793302.3793349>
- Miller 2025 study: <https://arxiv.org/pdf/2507.21280>
- Anderson 2025 study: <https://arxiv.org/html/2605.10702v1#S4>