

# The Future Engineer

Redefining Software Engineering in the Age of AI

# Who Am I?

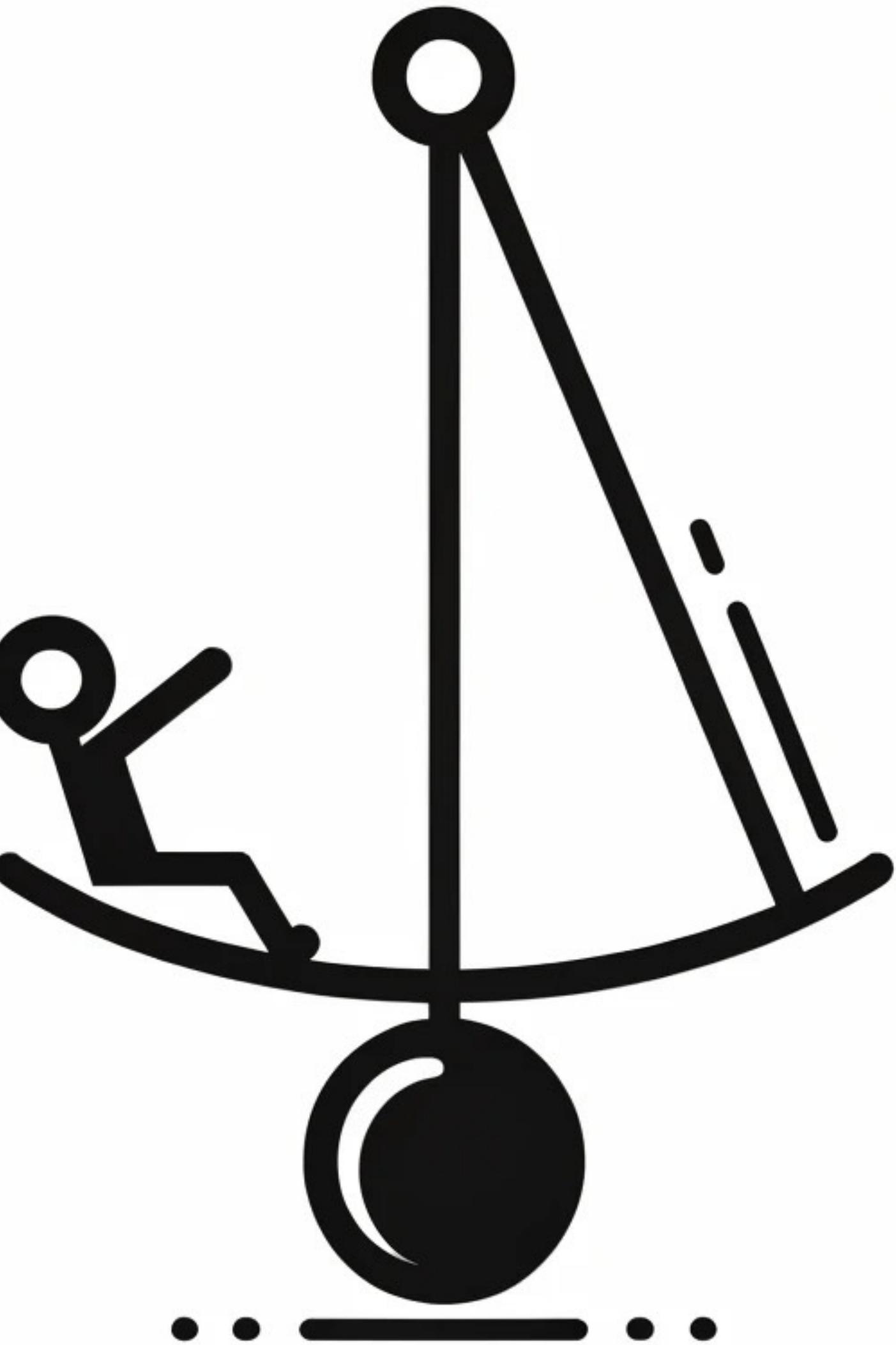
Software Engineer | Technical Leader | Self Professed Nerd

Present:

- Distinguished Engineer at Westpac NZ
- Master's of Engineering student at the University of Auckland

Previously:

- 2004 - 2014 - Software Engineer
- 2014 - Present - Technical Leader



Why is this topic so  
important?

*Software development is the  
invisible hand behind this  
digital magic, transforming  
ideas into the tools that shape  
our lives.*

<https://www.statista.com/outlook/tmo/software/worldwide>

# 28.7 million software engineers worldwide

US\$3,567 billion

<https://springsapps.com/knowledge/how-many-software-engineers-are-there-in-2024>



<https://x.com/KevinNaughtonJr/status/1796241418484334722>

# Is this the End of Software Engineering?

A screenshot of a Reddit post from the subreddit r/WGU. The post is titled "Is software engineering dying??". The author is BennieMiller and it was posted 1 year ago. The post content discusses concerns about the future of software engineering due to massive layoffs and AI automation. It has 94 upvotes and 133 comments. The URL of the post is [https://www.reddit.com/r/WGU/comments/1b89euw/is\\_software\\_engineering\\_dying/](https://www.reddit.com/r/WGU/comments/1b89euw/is_software_engineering_dying/).

A screenshot of a Hacker News post. The title is "Is it too late to enter software engineering in 2024?". The author is lottle and it was posted 11 months ago. The post content discusses concerns about the future of software engineering due to AI and the recession. It has 4 points and 13 comments. The URL of the post is <https://news.ycombinator.com/item?id=39664697>.

A screenshot of a Reddit post from the subreddit r/AskProgramming. The post is titled "My dad says I shouldn't do software development because those jobs will die in 10 years, is he right?". The author is Garlic\_Hacker and it was posted 2 years ago. The post content discusses concerns about the future of software development due to AI and oversaturation. It has 1 upvote and 1 comment. The URL of the post is [https://www.reddit.com/r/AskProgramming/comments/yqg0wk/my\\_dad\\_says\\_i\\_shouldnt\\_do\\_software\\_development/](https://www.reddit.com/r/AskProgramming/comments/yqg0wk/my_dad_says_i_shouldnt_do_software_development/).



# Study Design

- Longitudinal study on the **impact of AI on software engineering**
- Focusing on perceptions of professional software engineers using AI coding assistants
- Answering research questions on
  - Workflow and task focus
  - Productivity and developer experience
  - Skills acquisition and retention
  - Prompt engineering as a core competency
- Mixed methods approach
  - 2 x questionnaires
  - Semi-structured interviews

# Questionnaire #1

- Conducted online in October/November 2024 using Qualtrics
- Recruited participants through LinkedIn and other social and professional networks
- 33 organisations invited to participate worldwide, 9 accepted
- Ethics approval from the University of Auckland Human Participants Ethics Committee on 23/09/2024 for three years. Reference Number: UAHPEC27902.
- Key Stats:
  - Total Responses: 224
  - Completed Questionnaires: 192
  - Professional Software Engineers: 184
  - Professional Software Engineers Using AI: 168

# Insights from related research

# Productivity Improvements Vary

Developers using GitHub Copilot completed a JavaScript programming task **55.8% faster** than those without access to the tool

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. [arXiv:2302.06590](https://arxiv.org/abs/2302.06590).

When using GitHub Copilot, ANZ engineers saw a **42.36% improvement** in productivity on average

Chatterjee, S., Liu, C. L., Rowland, G., & Hogarth, T. (2024). The Impact of AI Tool on Engineering at ANZ Bank: An Empirical Study on GitHub Copilot within Corporate Environment. [arXiv:2402.05636](https://arxiv.org/abs/2402.05636)

Developers using GitHub Copilot reported up to a **50% reduction** in time spent on coding tasks, with an overall projected **productivity improvement of 33-36%** in cloud-first development

Pandey, R., Singh, P., Wei, R., & Shankar, S. (2024). Transforming Software Development: Evaluating the Efficiency and Challenges of GitHub Copilot in Real-World Projects. [arXiv:2406.17910](https://arxiv.org/abs/2406.17910).

Productivity measures the extent to which an individual feels effective and efficient in their work, creating value and achieving tasks.

*Definition of productivity,  
2024 Accelerate State of DevOps DORA Report*



The image shows the front cover of the '2024 Accelerate State of DevOps DORA Report'. The title 'DORA' is at the top in large, bold, black letters. Below it is a light blue rectangular area containing the year '2024' and the text 'Google Cloud'. The main title 'Accelerate State of DevOps' is centered in a large, bold, black font. A dark grey vertical bar on the left lists 'Gold Sponsors' with logos for Catchpoint, Chronosphere, DataDog, Deloitte, Excella, Gearset, liatrio, Middleware, and Ops Era. On the right, a large number '10' is displayed with the text 'A decade with DORA' underneath.

**DORA**

2024

Google Cloud

**Accelerate  
State of DevOps**

Gold Sponsors

catchpoint chronosphere DATADOG

Deloitte. Excella Gearset

liatrio Middleware OPS ERA

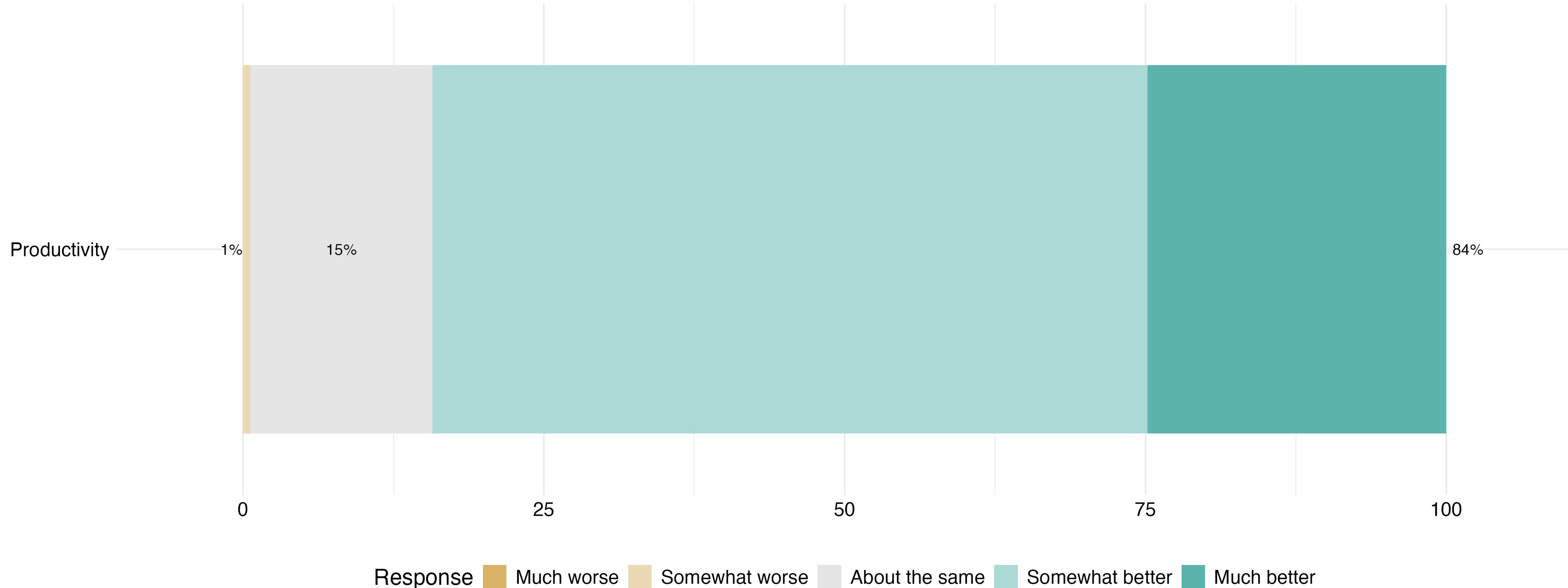
**10**

A decade with DORA

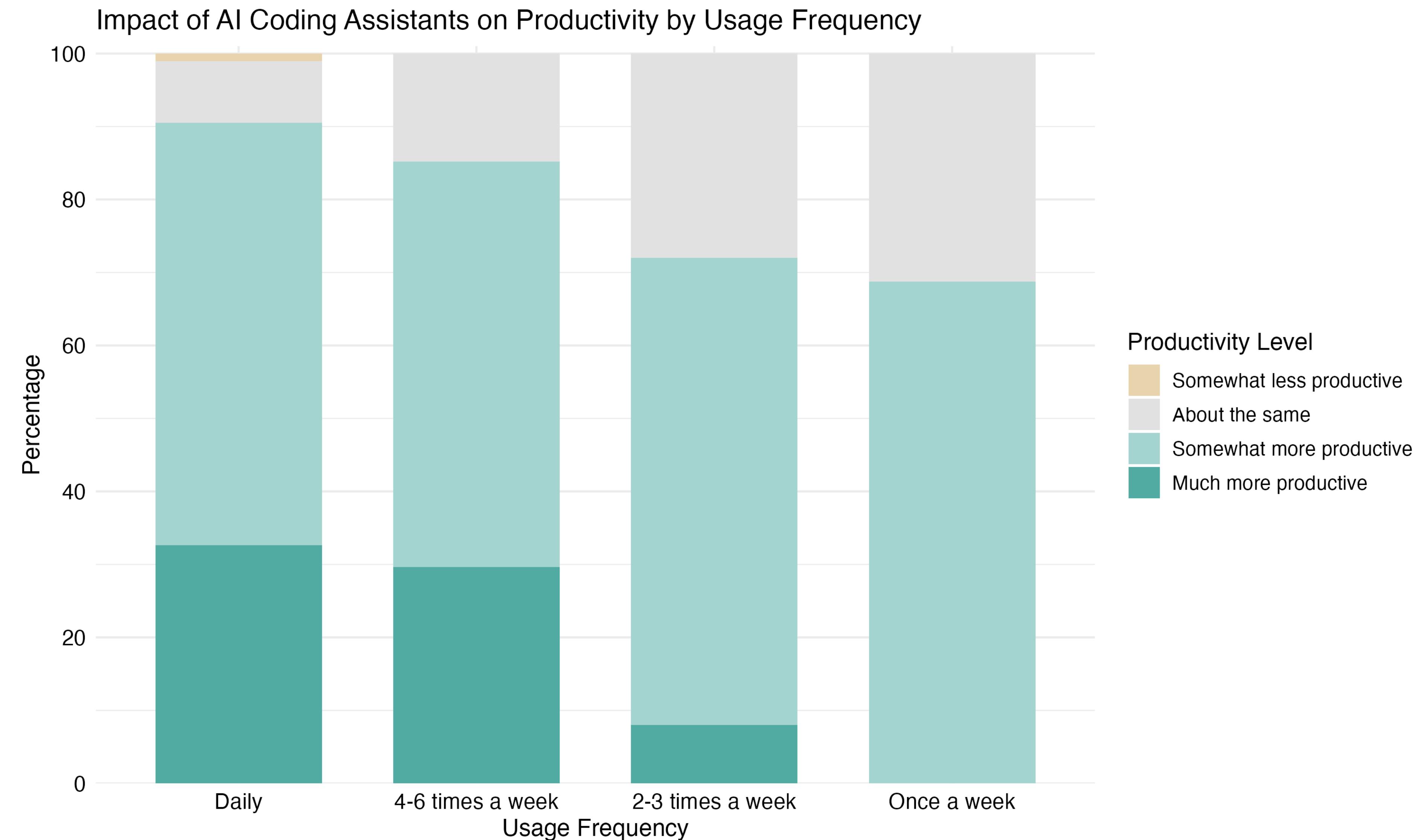
# What does our research say?

# 84% Report Improved Productivity

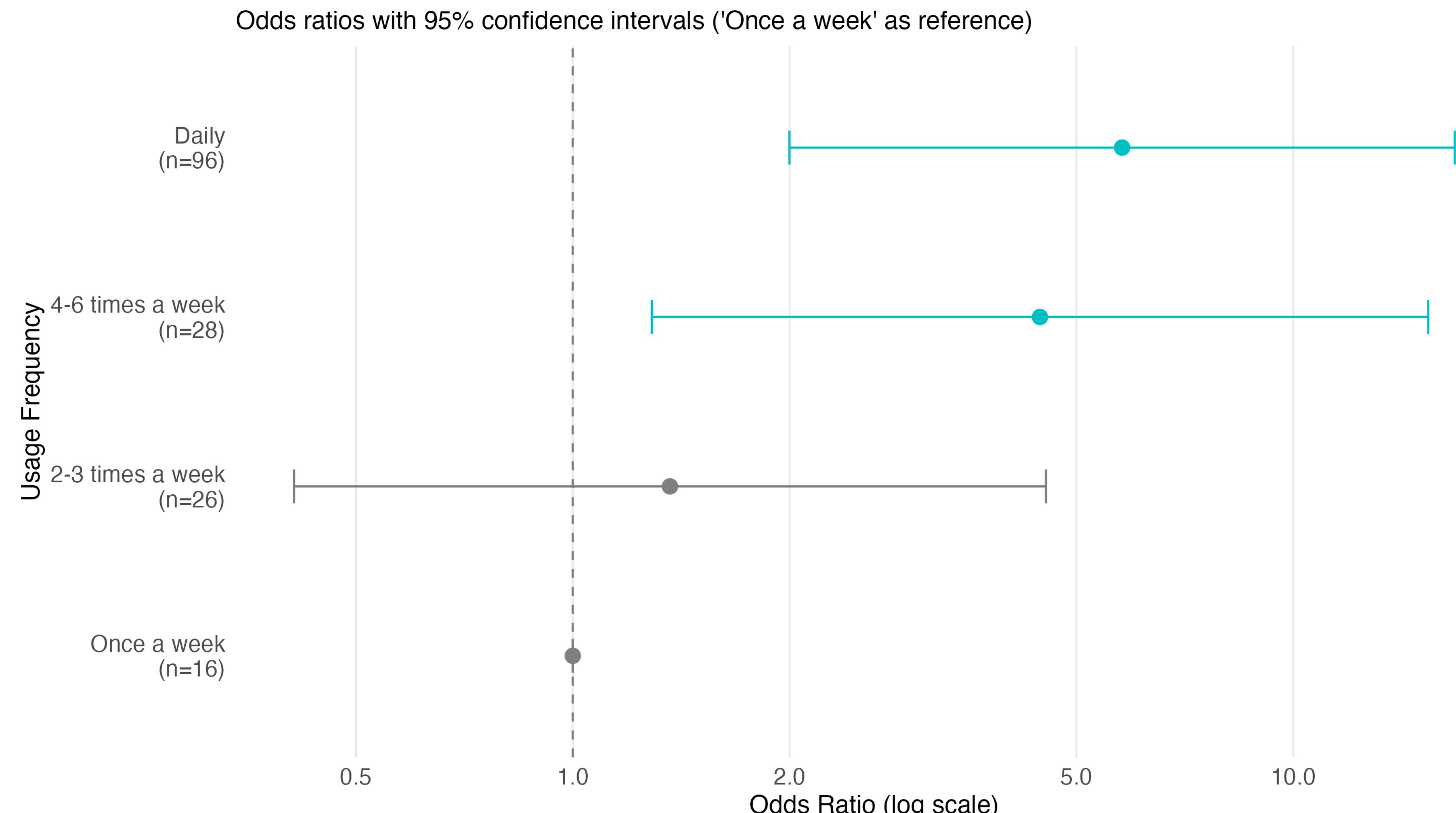
Impact of AI Coding Assistants on Productivity (Q4.1)



# How AI Usage Relates to Productivity



# Another View on Usage and Productivity



Note: Values  $> 1$  indicate the frequency is more likely to report improved productivity with AI assistants  
Values  $< 1$  indicate the frequency is less likely to report improved productivity  
Teal colors indicate statistically significant results ( $p < 0.05$ )

# More Usage = More Positive Sentiment

Frequent users mention:

 Faster development

 Less boilerplate code

 Simpler refactoring

 Faster prototyping

Infrequent users mention:

 Manual reviews and adjustments

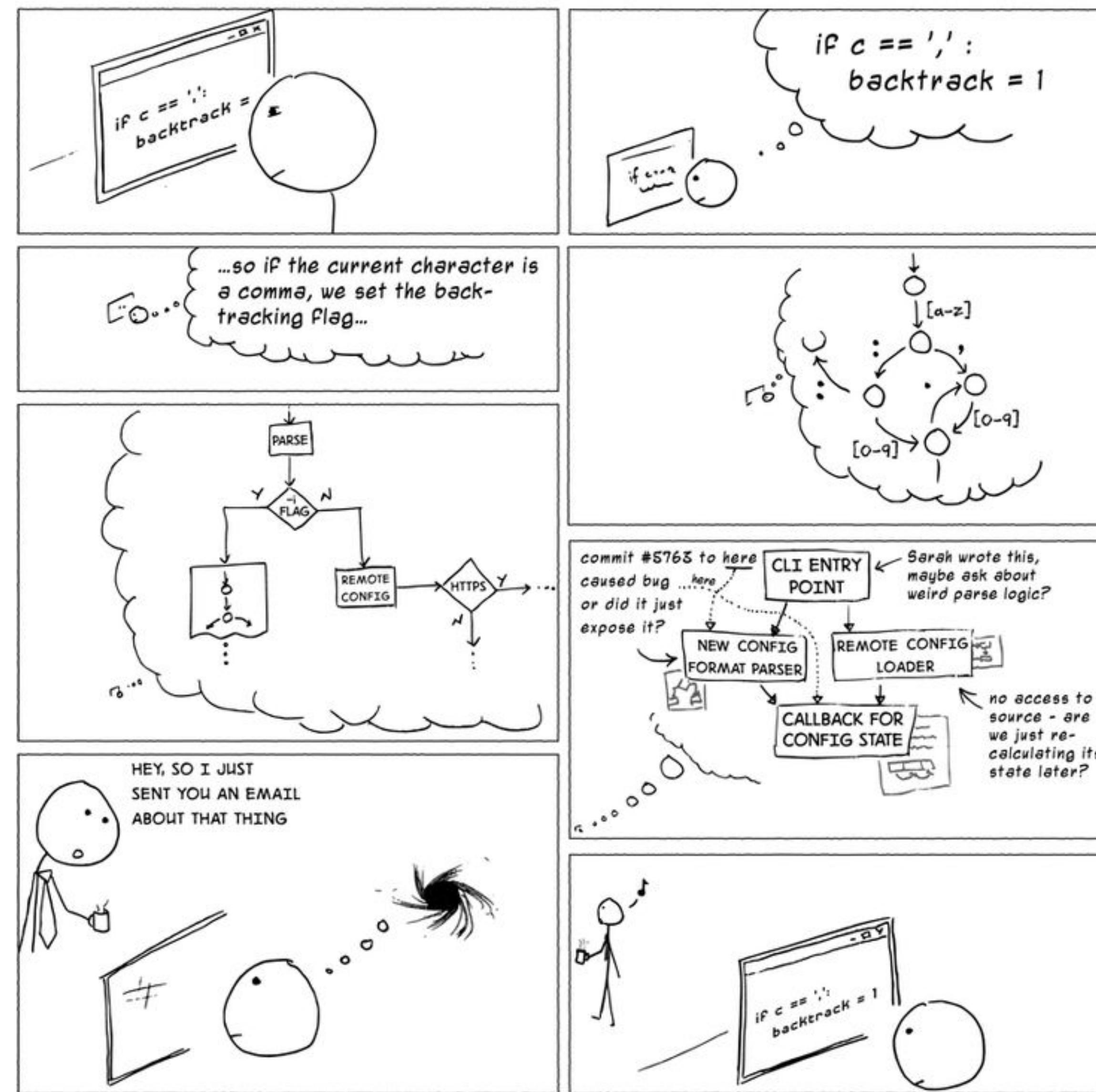
 Critical evaluation skills

 Technical exploration

 Faster development

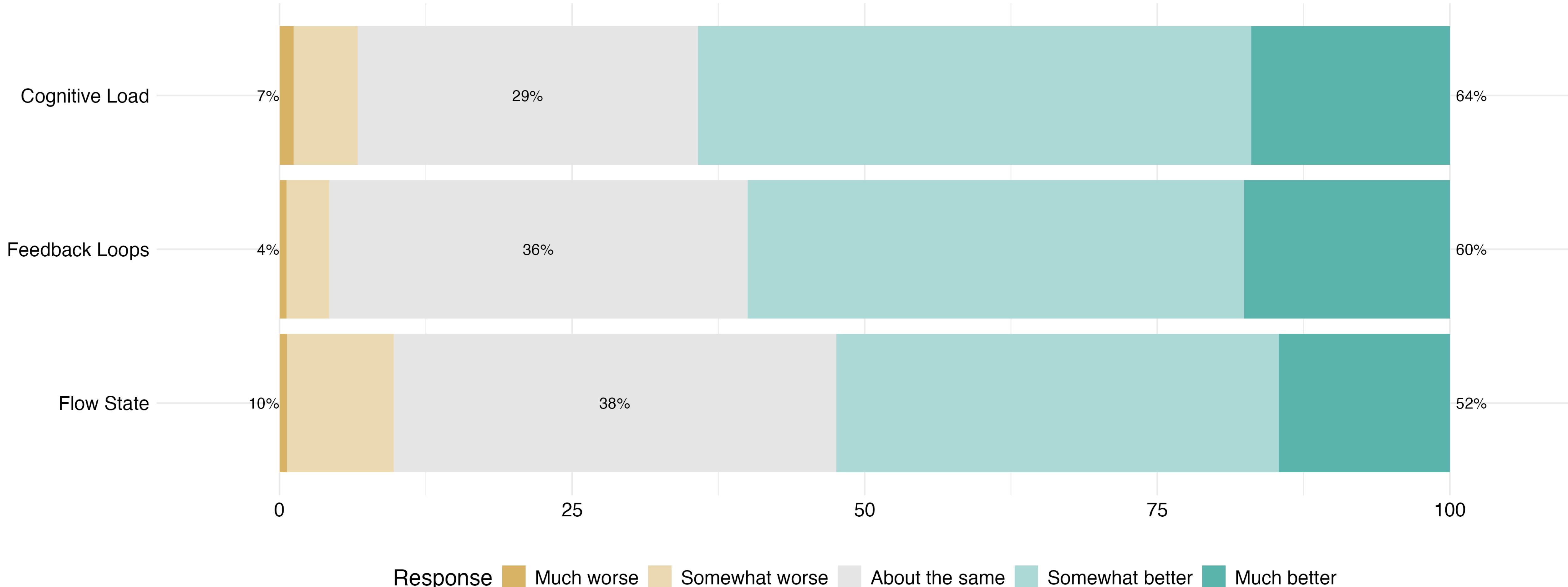
How does AI make us more  
productive?

# THIS IS WHY YOU SHOULDN'T INTERRUPT A PROGRAMMER

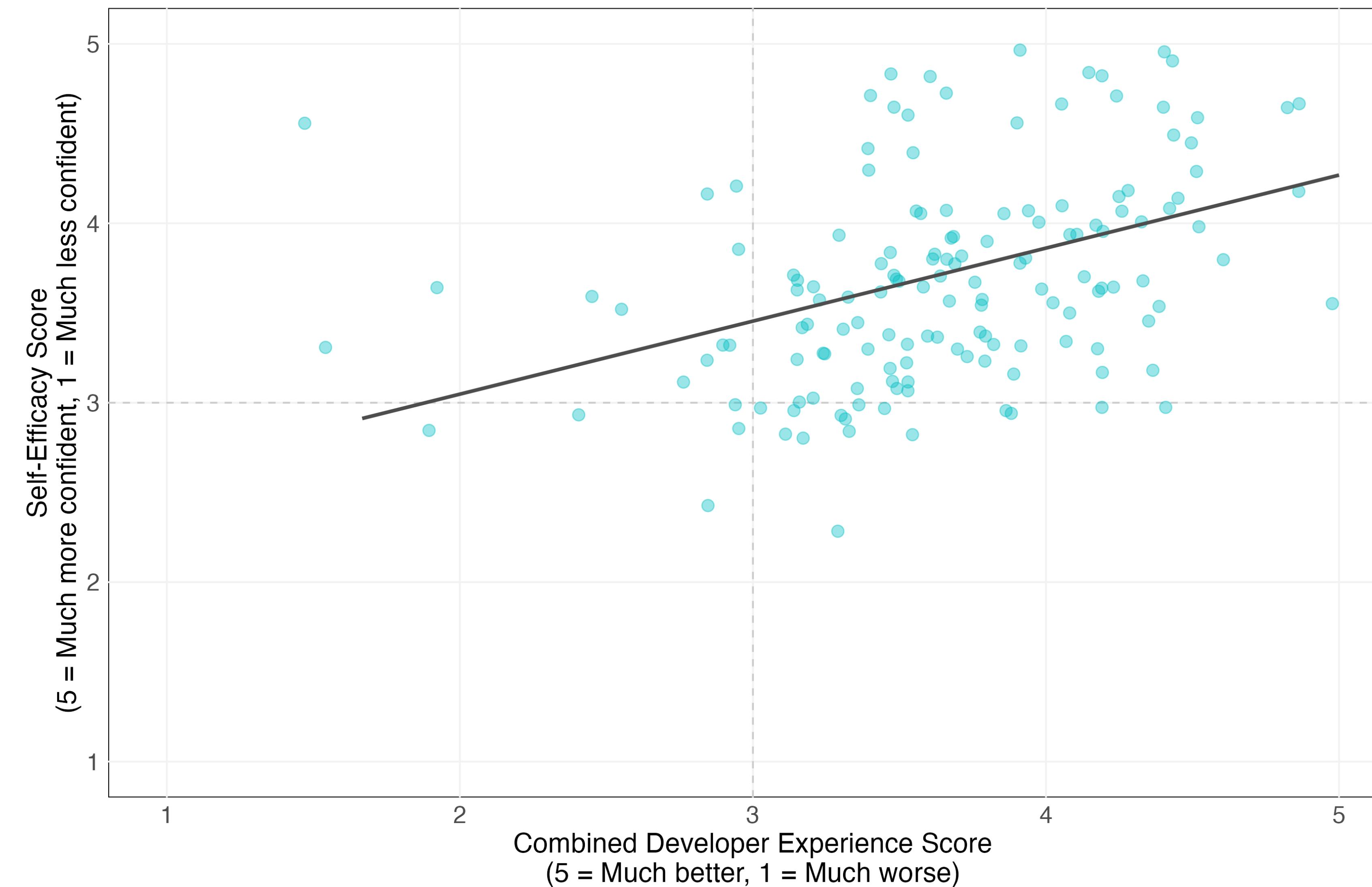


# Improved Developer Experience

Impact on Aspects of Developer Experience (Cognitive Load, Feedback Loops, Flow State) (Q4.2)

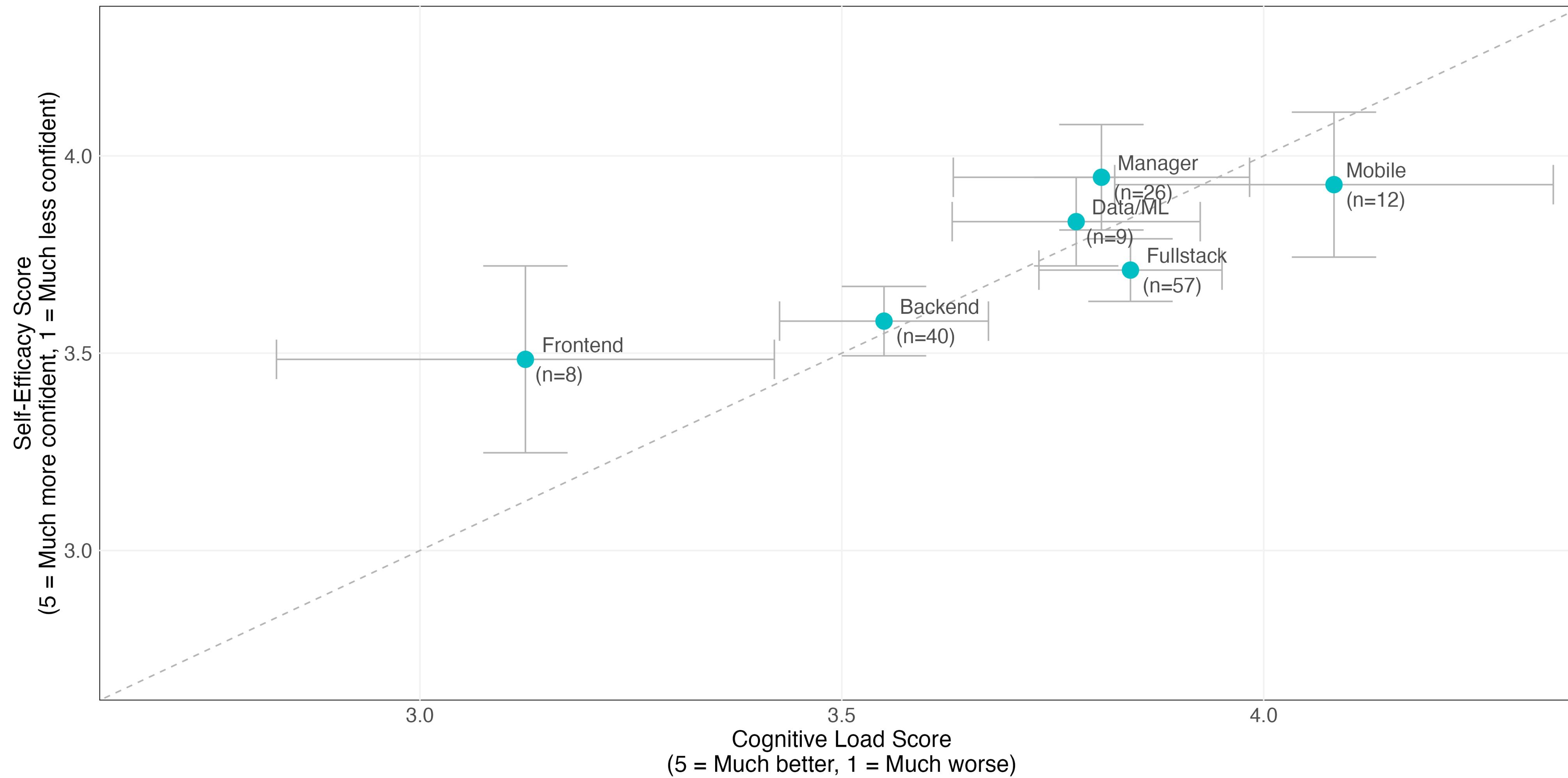


# Developer Experience & Self-Efficacy



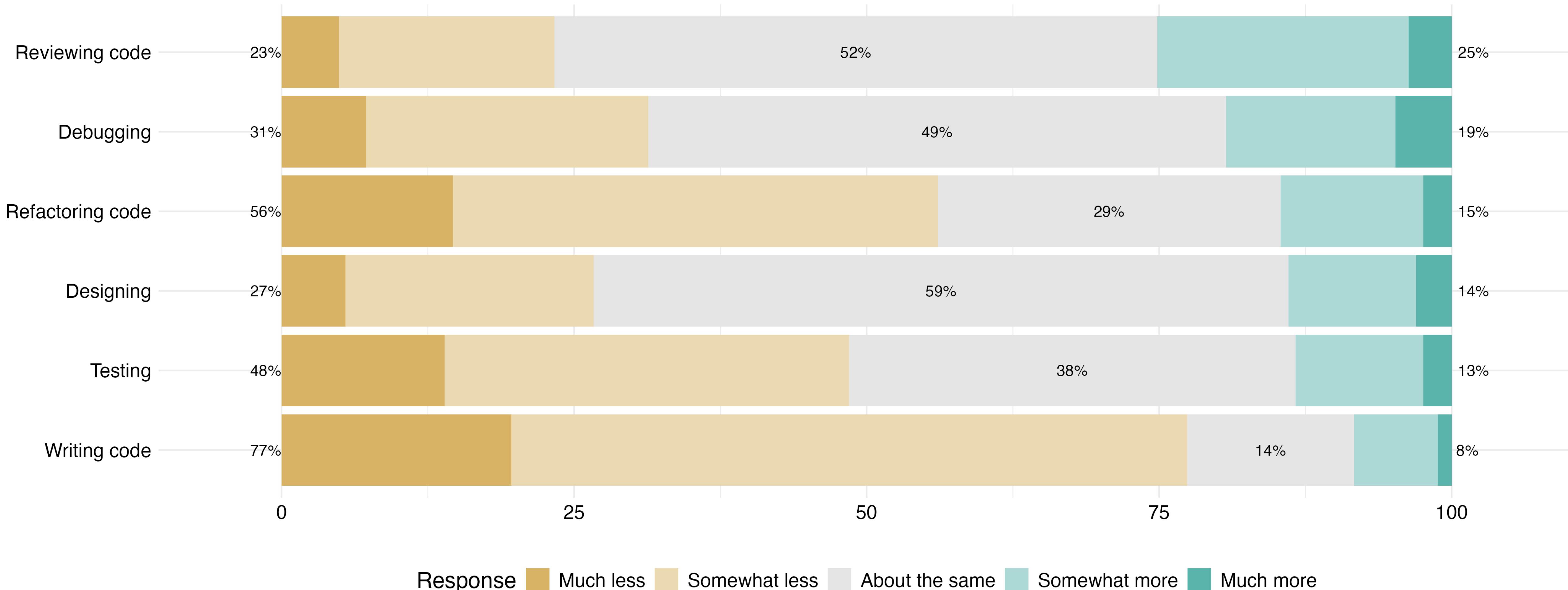
# Cognitive Load & Self-Efficacy by Role

Points above line: Greater improvement in self-efficacy  
Points below line: Greater improvement in cognitive load



# Shifts in Task Focus

Time Spent on Software Engineering Tasks with AI Assistance (Q3.4)





“Generating a bunch of boilerplate terraform is much easier now”

“I wouldn't have taken up a react front end role, AI assistant has enabled me to do it”

“When I am stuck this helps a lot. It's my coding buddy.”

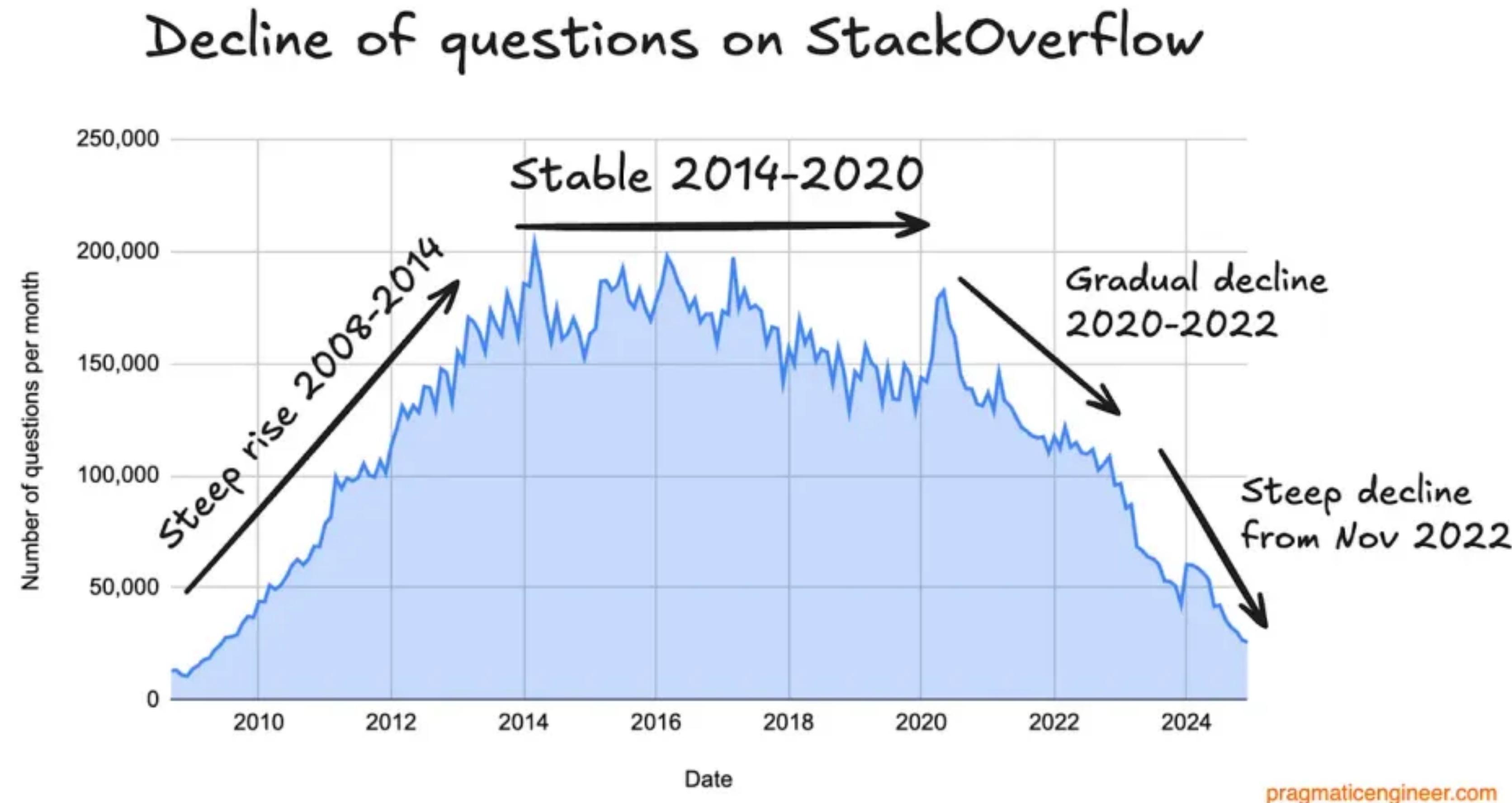


“Significantly helped me get new projects going”

“It can find solutions which would otherwise require search and experimentation”

“A situation whereby I had to integrate a technology I haven't used before, AI helped me effortlessly”

# Replacing Traditional Tools



<https://blog.pragmaticengineer.com/are-llms-making-stackoverflow-irrelevant/>

# Where AI Helps

## Time Efficiency

- Increased speed
- Faster boilerplate code
- Faster prototyping
- Easier test generation
- Get answers faster

## Cognitive Support

- Enables technical exploration
- Explains legacy or complex code
- Makes a good 'rubber ducky'
- Allows more focus on problem definition and system design

## Improved Flow State

- Replaces traditional tools (e.g. Google, StackOverflow, documentation)
- Keeps software engineers in the IDE

So where's the catch?

## Days before OpenAI

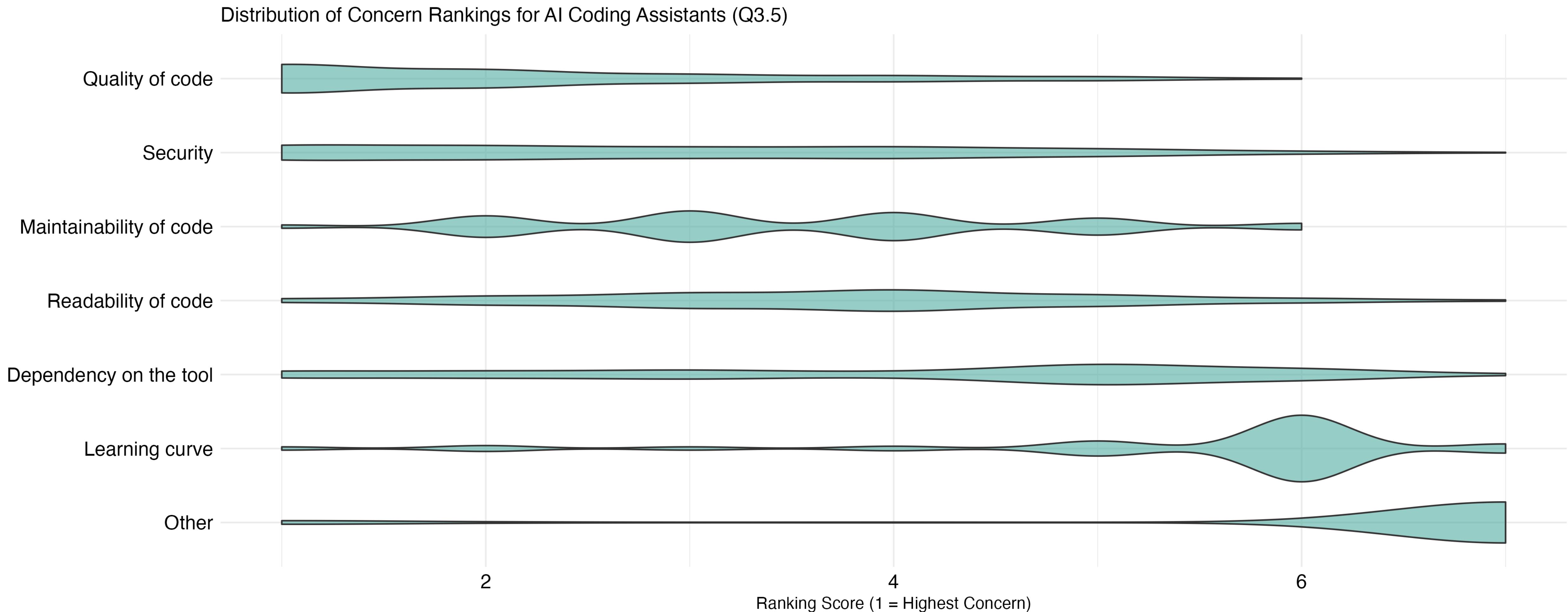


## Days after OpenAI



[https://www.reddit.com/r/meme/comments/14pm80n/chatgpt\\_is\\_actually\\_a\\_problem\\_for\\_developers/](https://www.reddit.com/r/meme/comments/14pm80n/chatgpt_is_actually_a_problem_for_developers/)

# Software Engineers Have Some Concerns





"I have found large proposals for whole objects generally poor quality in Swift, often including code which will not build, contains non-existents api functions or does build but is poorly implemented."

"The code it created was literally not functional code"

"Only time it has hindered me so far is when I was starting out when i didn't know how to construct prompts."

"I also realised how dependent I became on it for writing tests at work"

"Asking the AI to fix it, giving it the error messages, usually got me in a loop, where after some responses, it would give me the first version again..."

"Sometimes it is quite painful to make it do what I want"

# Where AI Hinders

## Code Quality & Manual Adjustment

- Low-quality output
- Non-existent APIs
- Outdated suggestions
- Effort to refine outputs

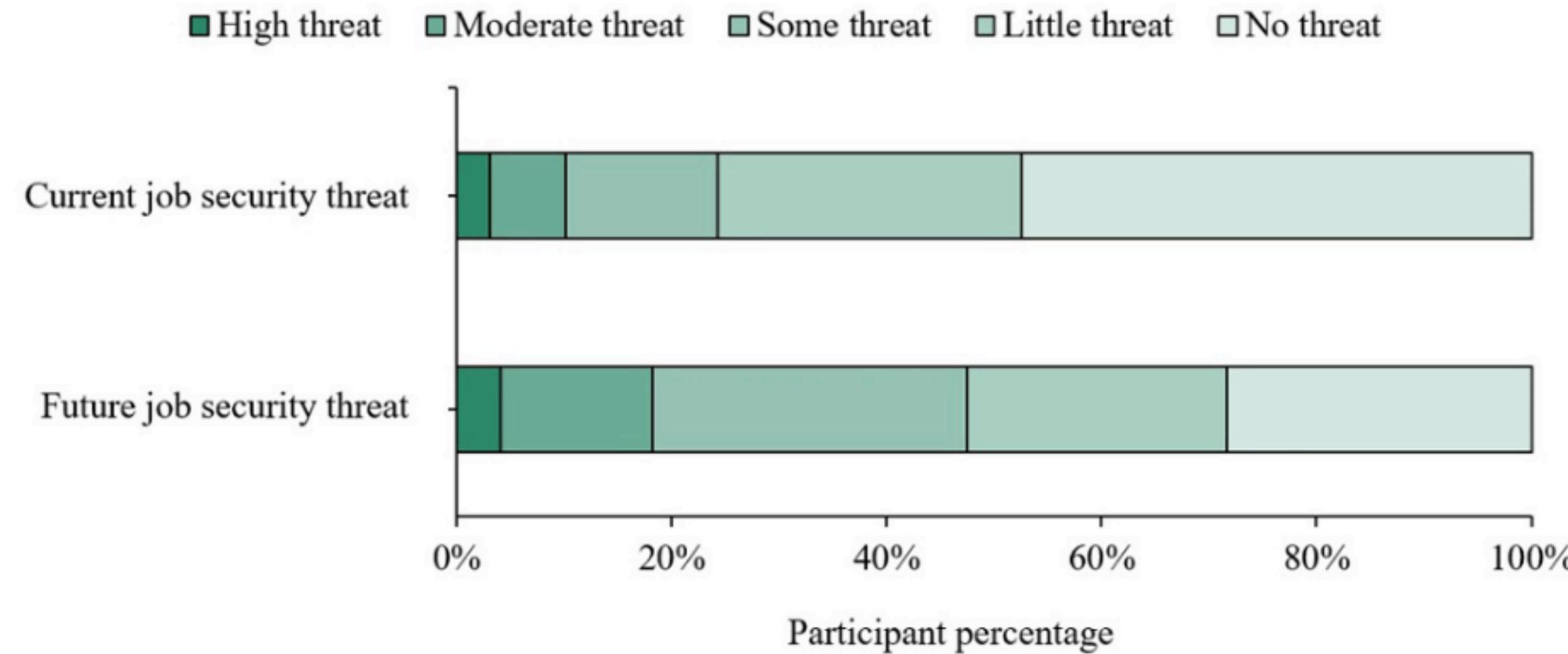
## Time Wasted & Frustration

- Prompt loops
- Ineffective solutions

## Job Displacement & Skill Erosion

- Dependency on AI
- Job security threat

# Related Research on Job Security Threat



**Fig. 4.** Participants' perceived current and future job security threats.

Kuhaila, M. A., Mathew, S. S., Khalil, A., Berengueres, J., & Shah, S. J. H. (2024). "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. [arXiv:2406.07765](https://arxiv.org/abs/2406.07765).

Pluralsight research finds 74% of software developers are planning to upskill in AI-assisted coding

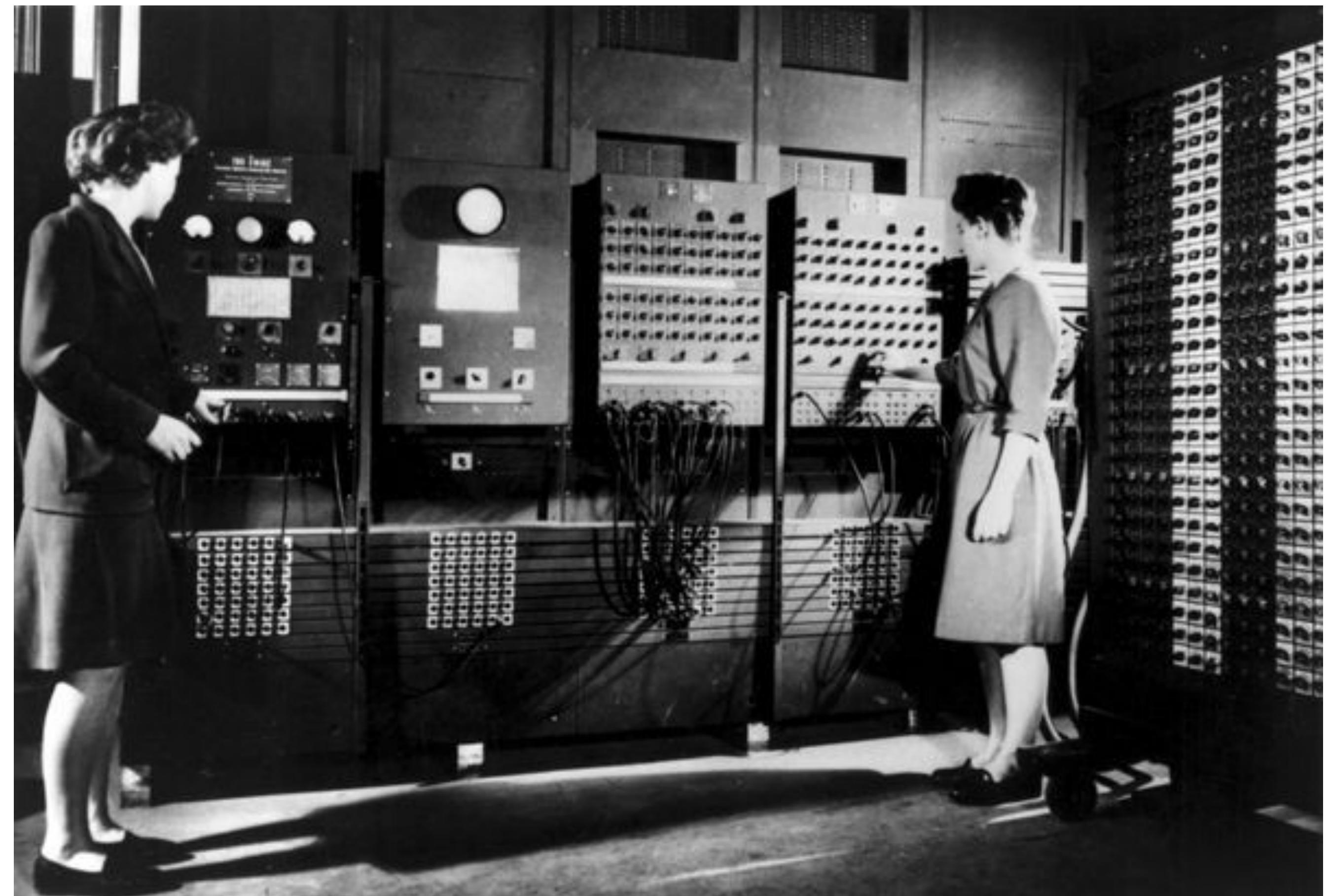
Oct 18, 2023

***Amid the rise of AI-assisted software development, the report also finds 45% of developers feel the value of their current skill sets are threatened.***

What does this mean for  
software engineers?

# The Role Isn't Disappearing, It's Evolving

- It's transforming into one focused on defining **intent** and **composing** solutions
- This isn't the first time:
  - Connecting physical circuits →
  - Writing binary code →
  - Writing assembly code →
  - Writing higher-level language code →
  - Writing specifications in natural language



Betty Jean Jennings and Frances Bilas program the ENIAC in 1946. [Computer History Museum](#)

# Core Principles Still Reign

## Software engineering fundamentals

- Foundational computer science concepts
- Strong software design principles
- Design patterns

## Critical thinking & verification

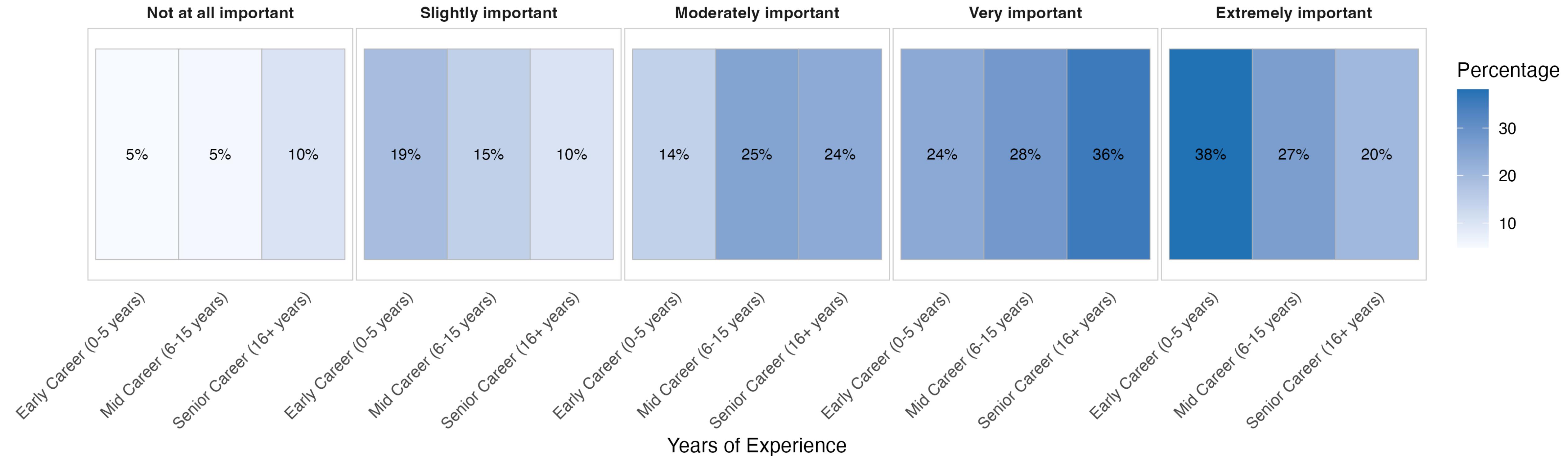
- Clear requirements
- Testing and code reviews
- Security risks

## Soft skills, curiosity & learning

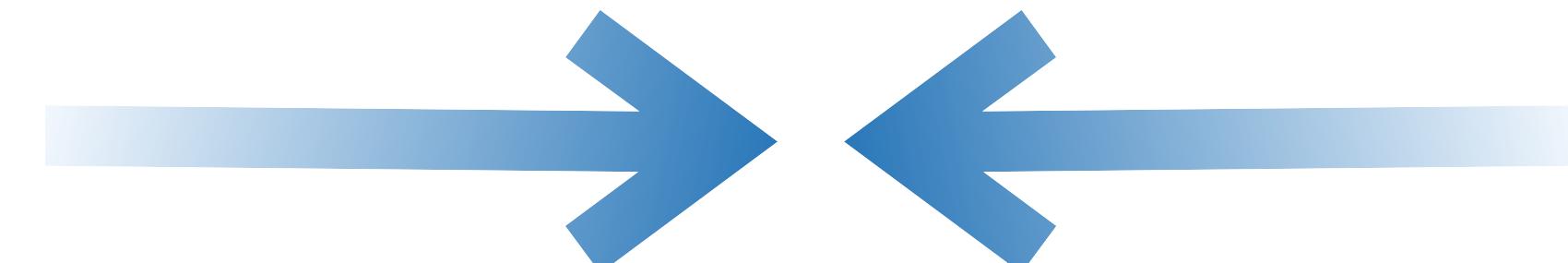
- Communication and collaboration
- Business acumen and problem context
- Human insight and creativity

# Prompt Engineering as a Natural Evolution

## Importance of Prompt Engineering Skills by Years of Experience (Grouped)



Those earlier in their careers value prompt engineering as a skill the most



What does this mean for  
engineering leadership?

# AI Mismanagement: A Cautionary Tale

- Mismanagement of AI leads to engineers feeling replaced, devalued and frustrated
- This creates low morale, high churn, and poor adoption of AI tools

A screenshot of a Reddit post from the subreddit r/webdev. The post, titled "Job is becoming unbearable due to AI", was made 10 days ago by user mace\_endar. The post discusses how the user's job is becoming unbearable due to their CEO's micro-managing, which involves passing code from GitHub repos into ChatGPT and then giving instructions to the team. The user notes that at least 50% of AI suggestions are hallucinations or not practical. The post has received 5.3K upvotes and 588 comments. There are also share and save buttons.

r/webdev • 10 days ago  
mace\_endar

## Job is becoming unbearable due to AI

My job is slowly becoming unbearable due to AI. If I didn't have any bills to pay or a family to support, I would have resigned already.

Our micro-managing CEO now regularly passes code from our GitHub repo into ChatGPT and then gets back to us with instructions on things that need to be changed or implemented in order to fix problems that we are dealing with. Usually accompanied by comments such as "why did you guys not come up with that already" or "just do it the same way as it's in my Python script, because it works".

The problem with it is that at least 50% of the suggestions are hallucinations, 40% aren't practical for us due to a variety of reasons (e.g. because AI suggests that we use Python libraries that haven't been updated in two years and we also don't use Python) and 10% are actually good and valid.

Add to that the now much tighter deadlines, because "with AI the coding takes care of itself now anyway", logging into Slack each morning seriously makes me want to puke.

5.3K 588 Share

[https://www.reddit.com/r/webdev/comments/1iixhh/job\\_is\\_becoming\\_unbearable\\_due\\_to\\_ai/](https://www.reddit.com/r/webdev/comments/1iixhh/job_is_becoming_unbearable_due_to_ai/)

# Human-AI Teaming - What Leaders Must Do

## Structured AI onboarding

- AI literacy training
- Build calibrated trust in AI by testing its limits in sandboxed environments
- Bridge AI acceleration with core engineering skills

## Evolving engineering roles

- Acknowledge AI changes how engineers work
- Shift from implementation to intent
- Architecture & system design
- Critical evaluation of AI outputs

## Redefining success metrics

- Traditional metrics don't reflect AI's impact (e.g LOC, velocity)
- Track AI's impact on quality and maintainability
- Measure how well engineers collaborate with AI tools

# The future of software engineering

# It's Time To Shift Software Engineering Left

- An opportunity to redefine the craft
- It's time for software engineering to **shift left**, designing systems and orchestrating the build
- Focus on making great products that solve real problems and makes customer happy
- Adapt the whole SDLC

“AI won't take your job. It's somebody using AI that will take your job.”

*Richard Baldwin, 2023 World Economic Forum Growth Summit*

# The Future of AI-Driven Software Engineering



Valerio Terragni, Annie Vella, Partha Roop, and Kelly Blincoe.  
2025. The Future of AI-Driven Software Engineering.  
ACM Transactions on Software Engineering and Methodology.  
Just Accepted (January 2025). <https://doi.org/10.1145/3715003>

# The Future Engineer: Adapting, Evolving, Thriving

- AI isn't replacing software engineers - it's redefining what software engineering is
- Our role is evolving - we're becoming system thinkers, AI collaborators, and problem solvers at a higher level
- In summary, **yes**, AI is having a large impact on software engineering and we must **adapt**



# Contact Details



Website: <https://annievella.com>

LinkedIn: <https://www.linkedin.com/in/annievella/>

Twitter/X: <https://x.com/codefrenzy>