Build Stuff, 13-15 Nov 2024
Vilnius, Lithuania

Become a 10x developer by harnessing the power of Generative AI

Ioannis Kolaxis
Director at Accenture

Ice Distribution

1918, USA

When electrical refrigerators were invented, ice distributors went out of job.

Will the same thing happen to developers due to Gen AI?

Image source: https://catalog.archives.gov/id/533758

Become a 10x developer by harnessing Gen AI

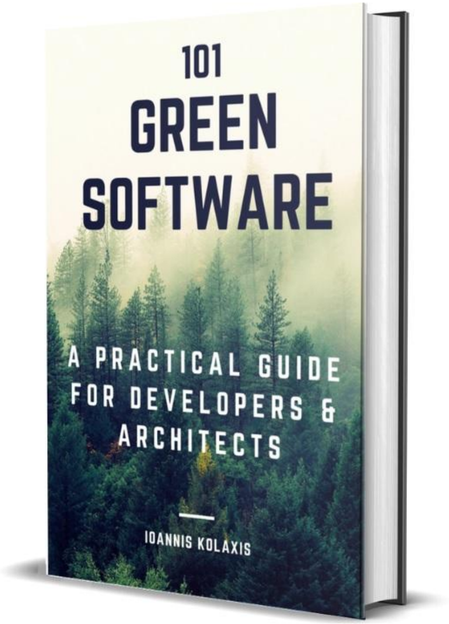
What will you learn from this session?

1. Weaknesses of AI-assisted coding
2. Strengths of AI-assisted coding
3. The future of software development

Who am I?

Ioannis Kolaxis - Director at Accenture

- Developer
  - From programming in BASIC on an Amstrad CPC 6128
  - To first job building Java Applets (in Java 1.2)
- Inventor: [5 Patents]
- Book author: Green Software



Visit website: kolaxis.dev

Do you use AI coding assistants?

GitHub Copilot, Amazon CodeWhisperer, ChatGPT, ...

Weaknesses of AI-Assisted Coding

1 Generating new code that is functionally incorrect

2 Refactoring existing code will likely break the code

3 They are negatively impacting code quality

A prompt example

```
/**
 * Check if in given list of numbers, are any two numbers
 * closer to each other than given threshold.
 *
 * Examples:
 * hasCloseElements([1.0, 2.0, 3.0], 0.5) returns false
 * hasCloseElements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3) returns true
 *
 * @param numbers   The list of numbers to check
 * @param threshold The threshold to compare the difference between numbers
 * @return true if any two numbers are closer to each other than the given threshold,
 *         false otherwise
 */
public static boolean hasCloseElements(List<Double> numbers, double threshold) {
}
```

Demo #1

Weakness #1

Generating new code that is functionally incorrect

Research:

- Generated code for 164 problems (from HumanEval dataset)
- Ran unit tests to check if generated code was functionally correct

Code Generation Tools	Correct Code (% problems)
ChatGPT (GPT-3.5)	65.2%
GitHub Copilot	46.3%
Amazon CodeWhisperer	31.1%

Source: Burak Yetiştiren, Işık Özsoy, Miray Ayerdem, and Eray Tüzün. 2023. Evaluating the code quality of AI-Assisted Code Generation Tools: An empirical study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. https://doi.org/10.48550/arXiv.2304.10778

Longer and more complex prompts generated functionally incorrect code

Prompts with simpler instructions generated correct code

Source: Burak Yetiştiren, Işık Özsoy, Miray Ayerdem, and Eray Tüzün. 2023. Evaluating the code quality of AI-Assisted Code Generation Tools: An empirical study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. https://doi.org/10.48550/arXiv.2304.10778

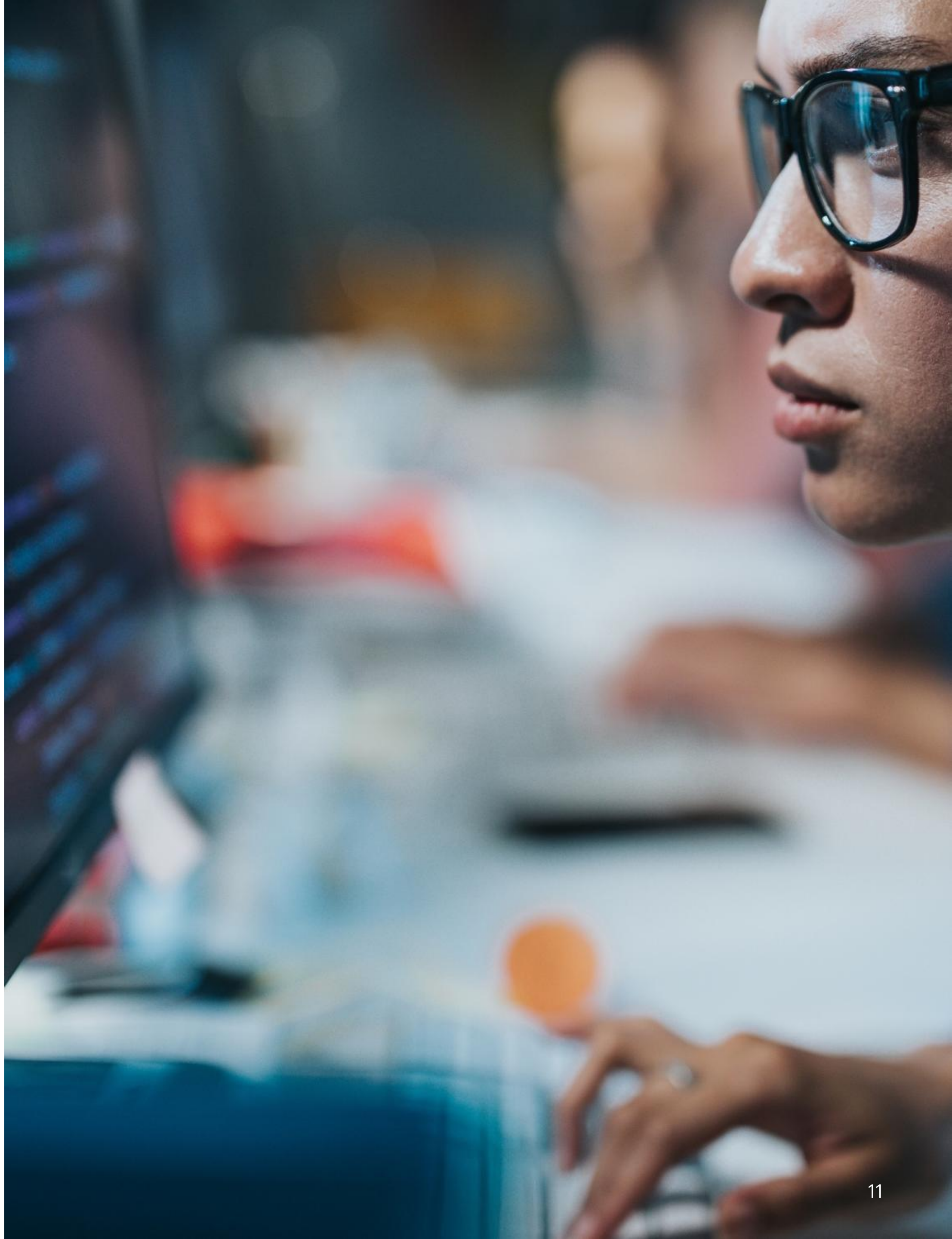
10x developers

Break big, complex problems into smaller, simpler ones →

Write methods for smaller problems

Write code prompts that have:

- Clear & accurate problem description
- Sample unit tests in Javadoc
- Descriptive method names



Weakness #2

Refactoring will likely generate incorrect code

Research:

- Refactored more than 100,000 real-world code smells in JavaScript & Typescript
- Ran unit tests to check if refactored code was functionally correct

AI Model	Correct Code Refactoring
PaLM 2 Code	37.29%
GPT-3.5	30.26%
Phind-CodeLlama-34B-v2	18.14%

Source: A. Tornhill, M. Borg, and E. Mones, 2024, “Refactoring vs Refuctoring: Advancing the state of AI-automated code improvements”
https://codescene.com/whitepapers

Common failures in AI-assisted refactoring

Dropped entire branches

- "if" blocks disappeared

Inverted boolean logic

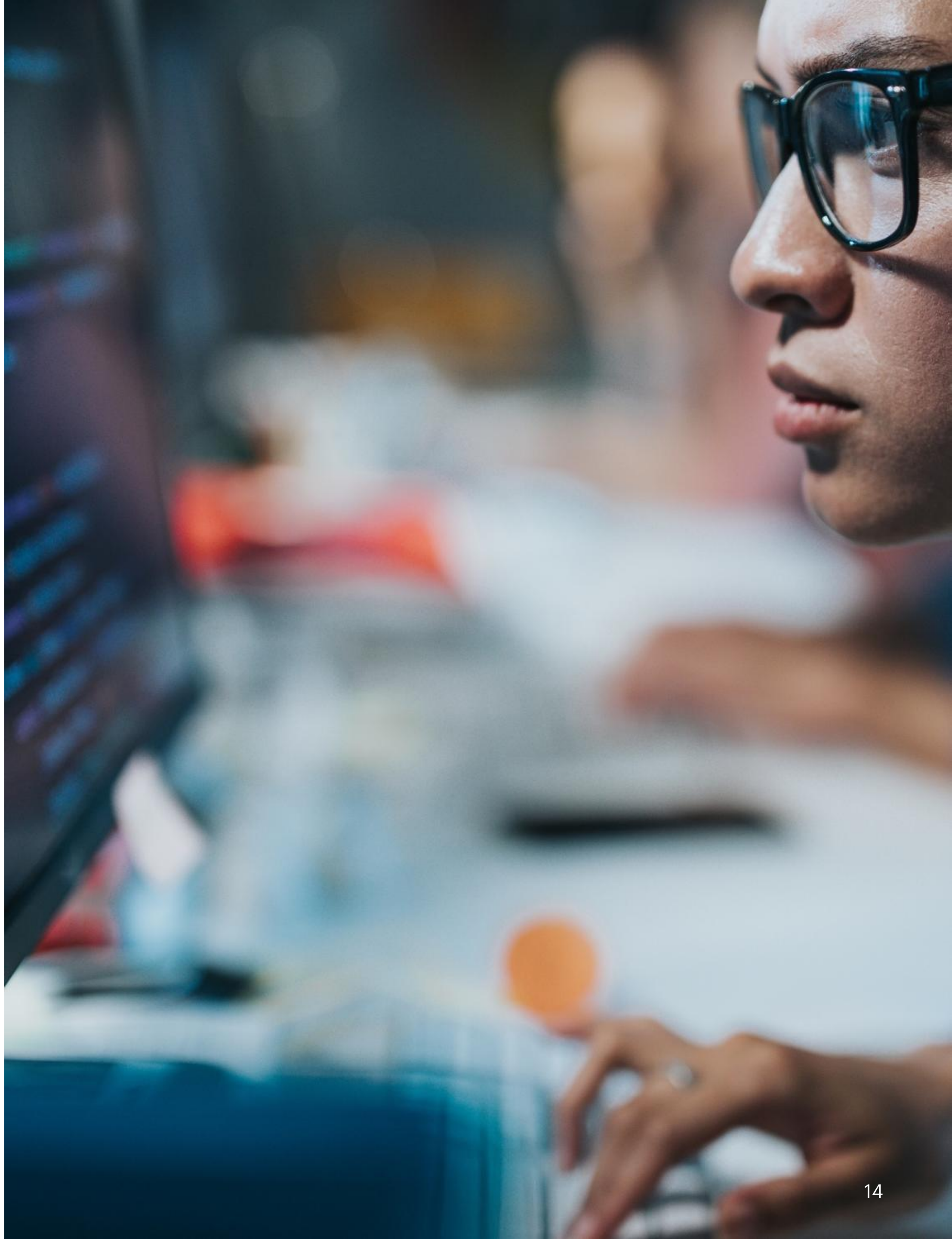
- Changed (a && b) to !(a && b)

Source: A. Tornhill, M. Borg, and E. Mones, 2024, "Refactoring vs Refuctoring: Advancing the state of AI-automated code improvements" https://codescene.com/whitepapers

10x developers

Verify that AI-refactored code is correct by:

- Running unit tests
- Reviewing code



As a developer, where do you spend most of your time?

A. Writing new code
B. Reading existing code
C. Waiting for a full build to complete
D. Other

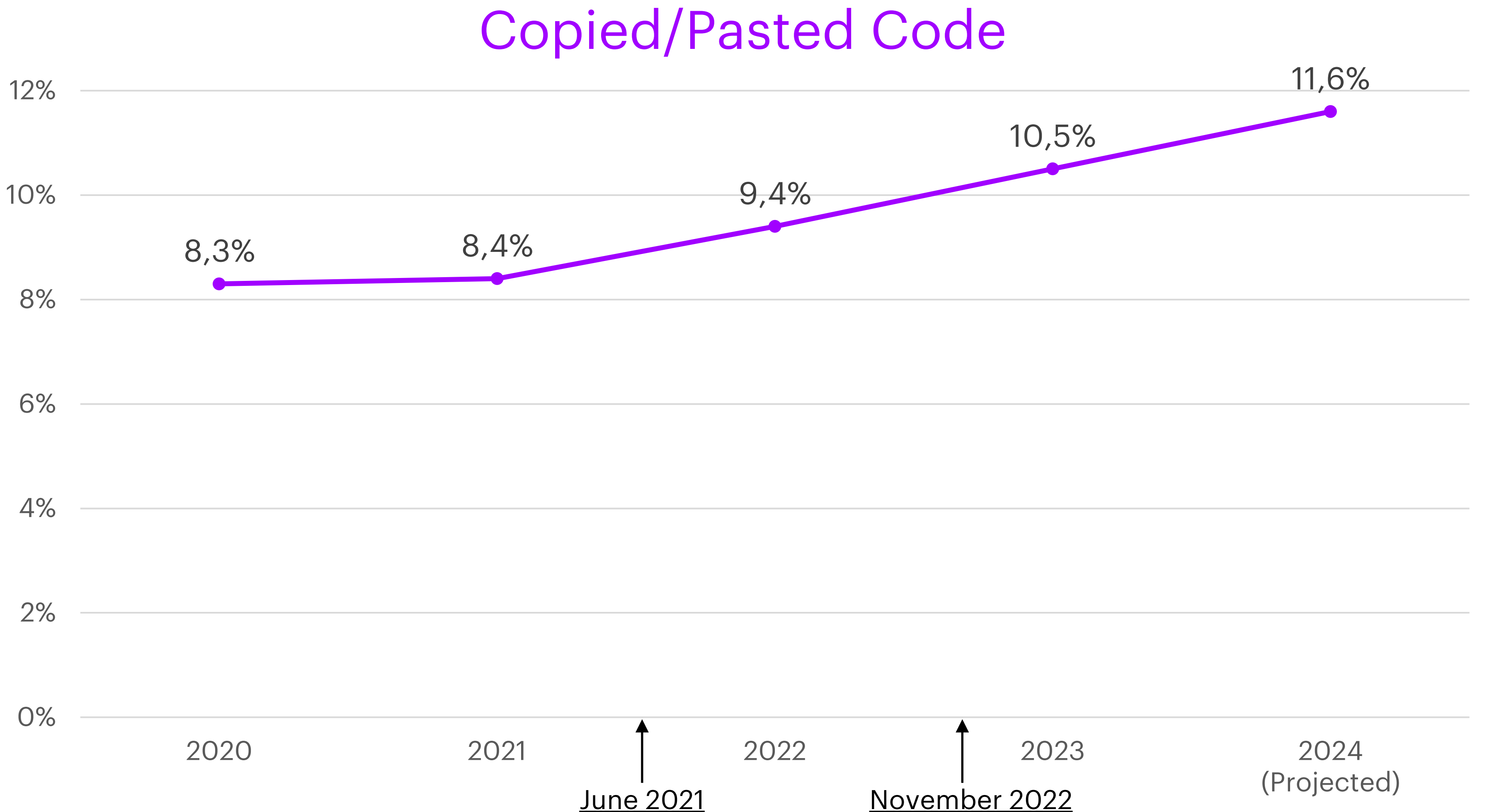
We spend 70% of our time reading code!

… and only 5% writing code

Source: R. Minelli, A. Mocci, and M. Lanza, 2015, “I Know What You Did Last Summer – An Investigation of How Developers Spend Their Time” https://ieeexplore.ieee.org/document/7181430

Weakness #3

AI-assisted coding makes it harder to maintain code



June 2021

GitHub Copilot

Technical Preview

November 2022

ChatGPT

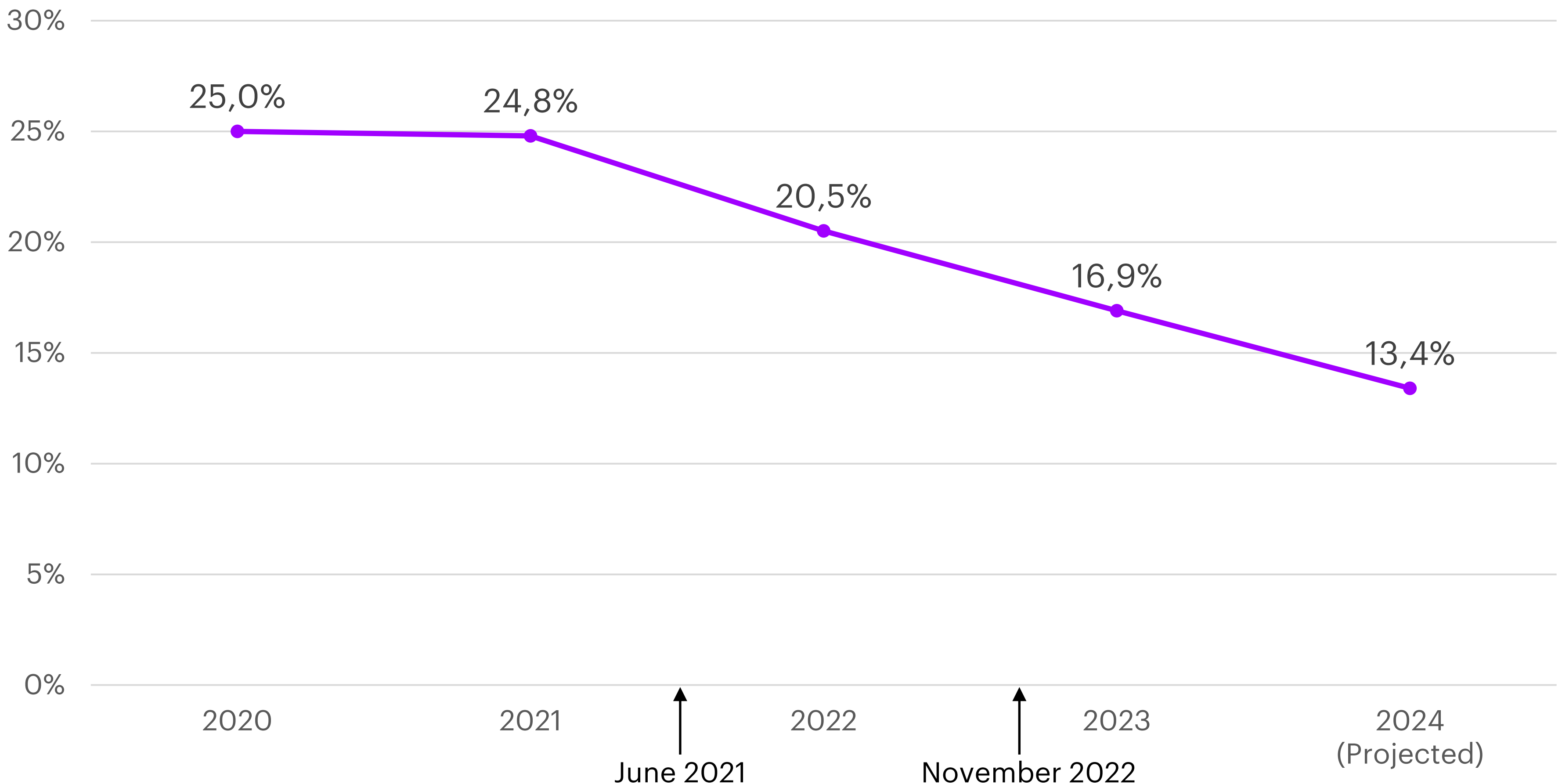
Initial Release

Source: W. Harding, and M. Kloster, 2024, “Coding on Copilot”, GitClear

Weakness #3

AI-assisted coding makes it harder to maintain code

Reused & Refactored Code



June 2021

GitHub Copilot

Technical Preview

November 2022

ChatGPT

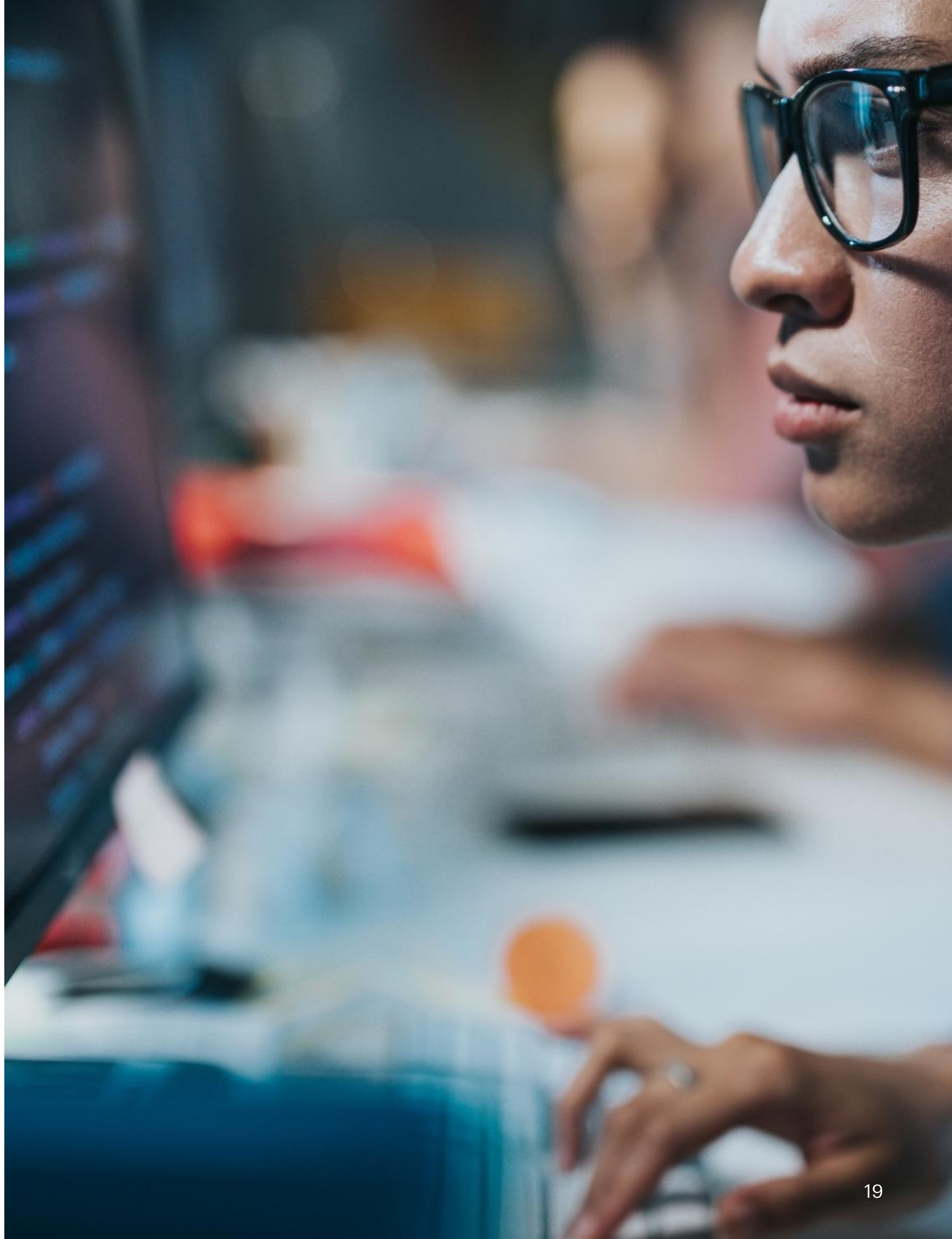
Initial Release

Source: W. Harding, and M. Kloster, 2024, “Coding on Copilot”, GitClear

10x developers

Prefer reusing existing code than copying/pasting via AI assistants

... that's why they can read & maintain code 10x faster

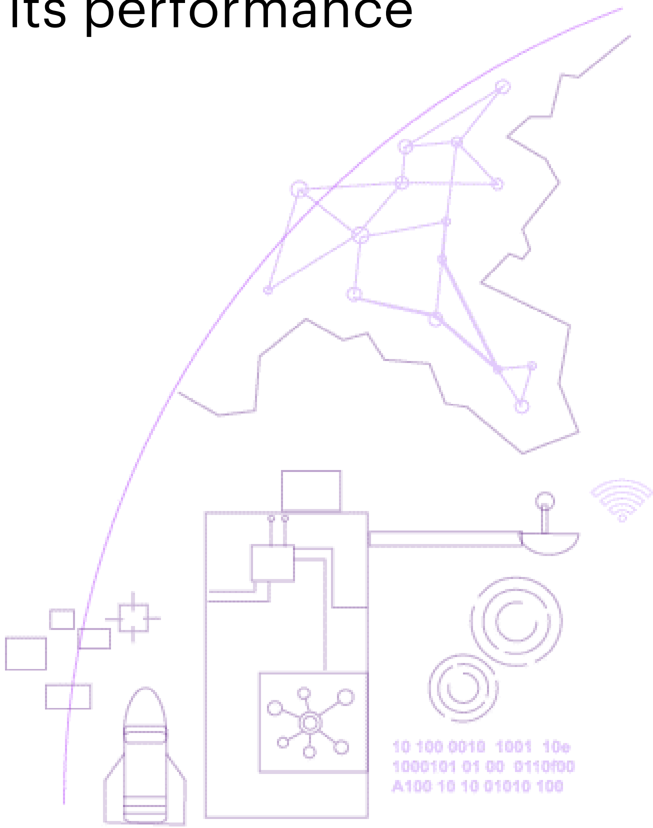


Strengths of AI-Assisted Coding

1 Exploring new technologies & developing prototypes

2 Understanding code & enhancing its performance

3 Generating unit tests



Strength #1

Exploring new technologies & developing prototypes

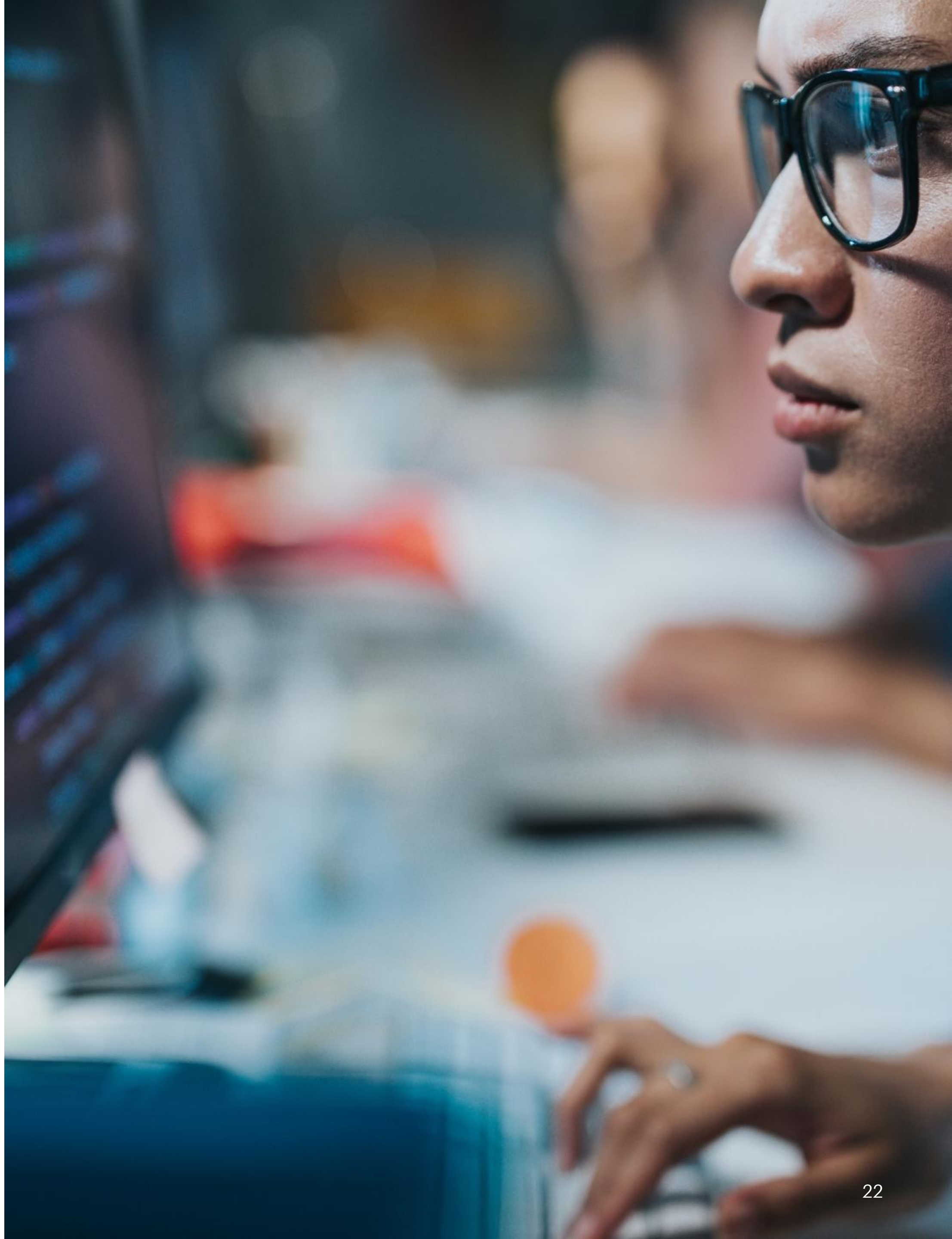
Build throw-away prototypes faster:

- Helps use unfamiliar APIs
- Keeps me focused when writing code

10x developers

Use AI-coding assistants to

move fast and break things



Strength #2:

Understanding code & enhancing its performance

Can you recognize this algorithm?

```
public static void sort(int[] array) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (array[j] > array[j + 1]) {
                int temp = array[j];
                array[j] = array[j + 1];
                array[j + 1] = temp;
            }
        }
    }
}
```

Demo #2

Refactoring code to use efficient algorithms

Experiment:

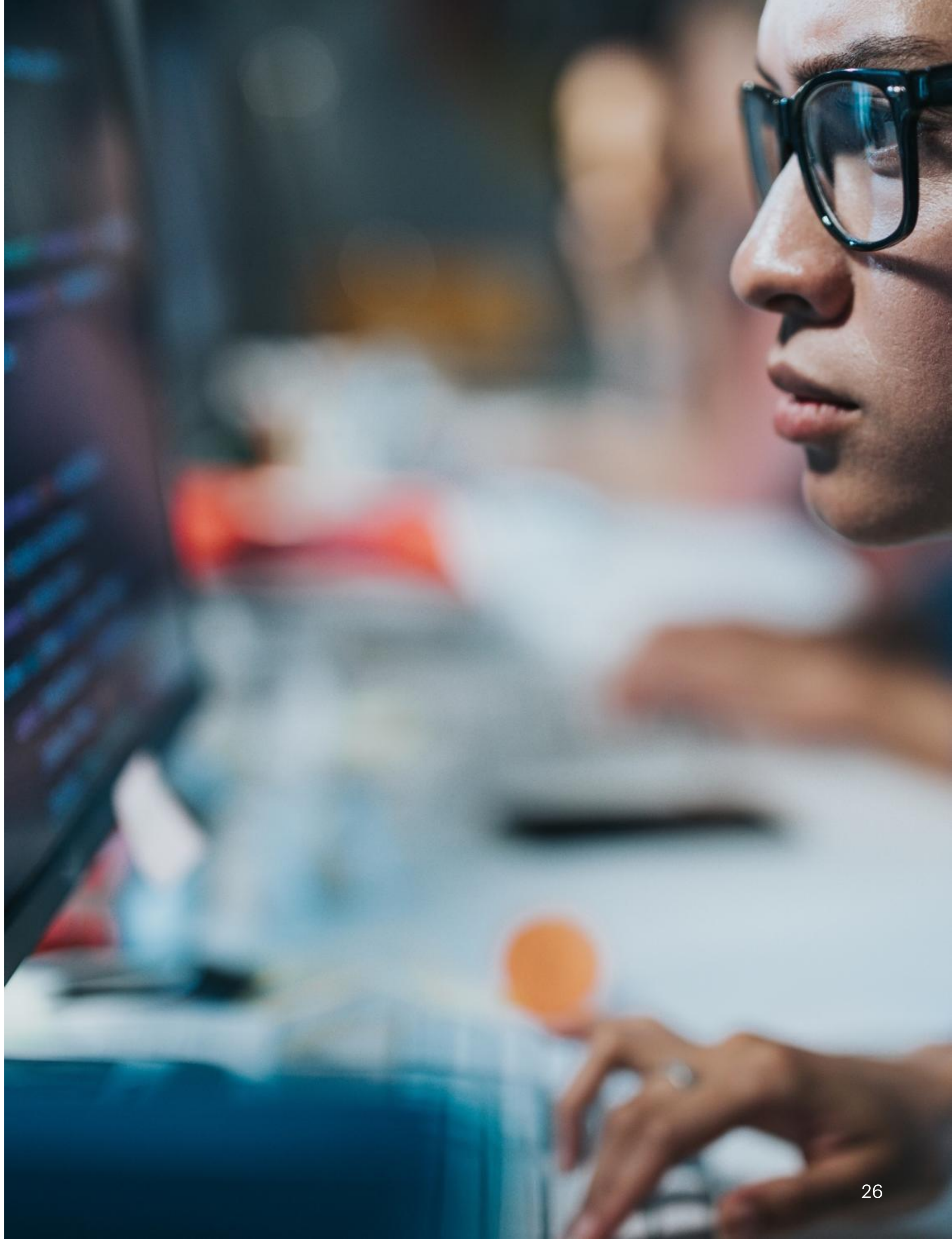
- Measured execution time to sort an array of 1,000 numbers.
- Refactored code (from Bubble Sort to Quick Sort) performs 95.4% faster

Algorithm	Execution Time (μsec)	Improvement (%)
Bubble Sort	1,249.1	
Quick Sort	57.5	**95.4%**

10x developers

Use AI assistants to understand code & identify opportunities to improve its performance

… that’s why their code runs 10x faster



Strength #3

Generating unit tests

Unit tests

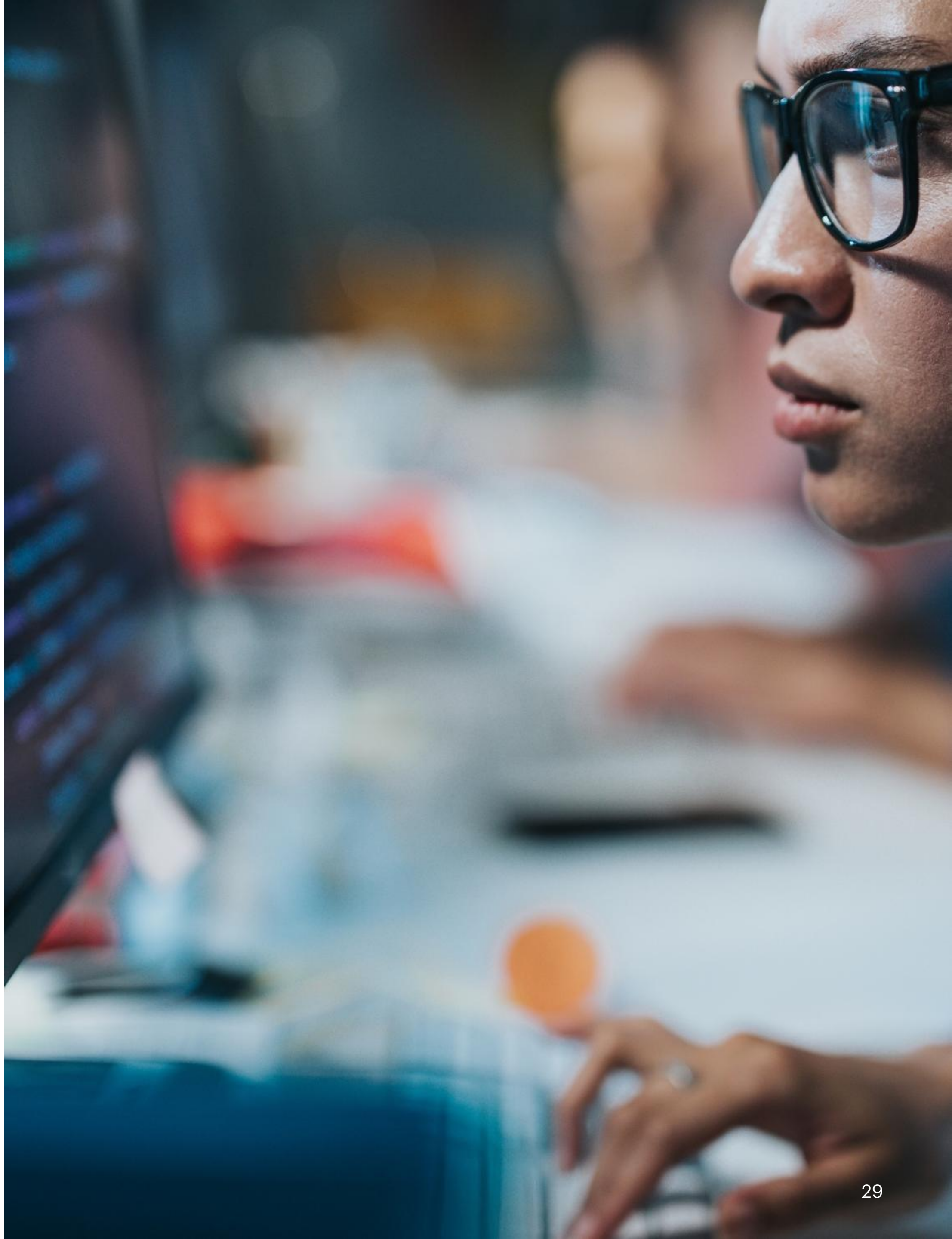
A safety net for software apps

Demo #3

10x developers

Move fast and don’t break things

… using AI coding assistants to generate unit tests



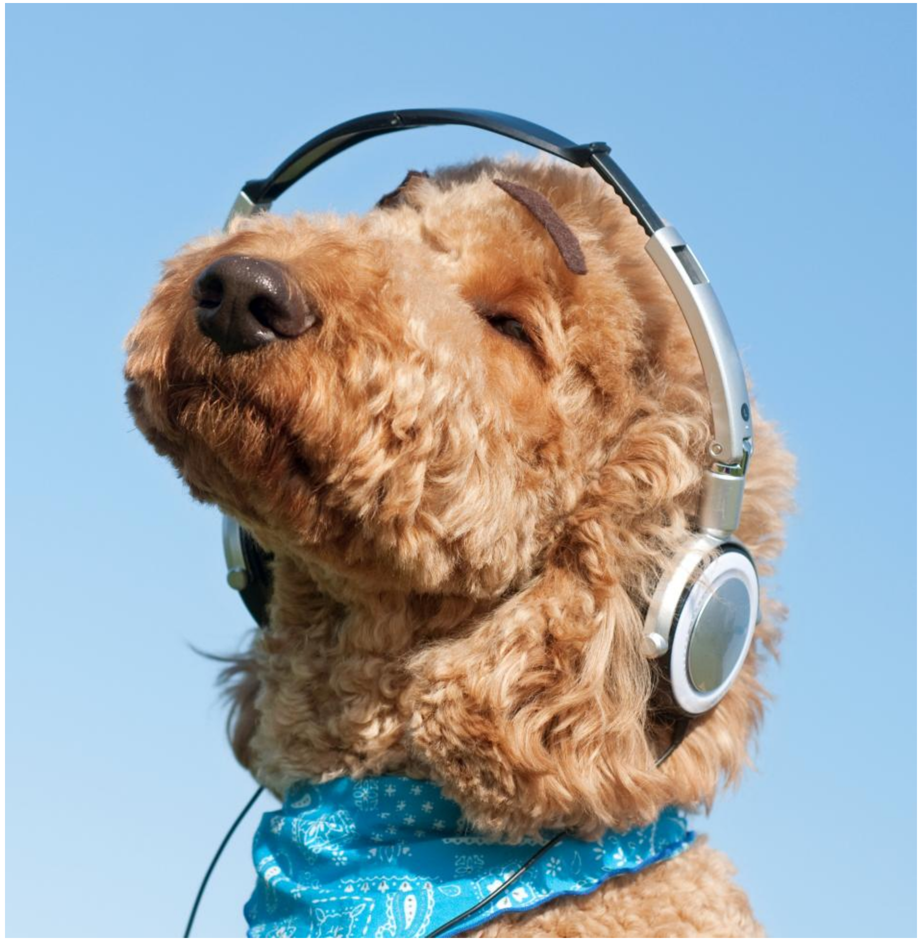
What is the future of software development?

Image created using Midjourney

Demo #4

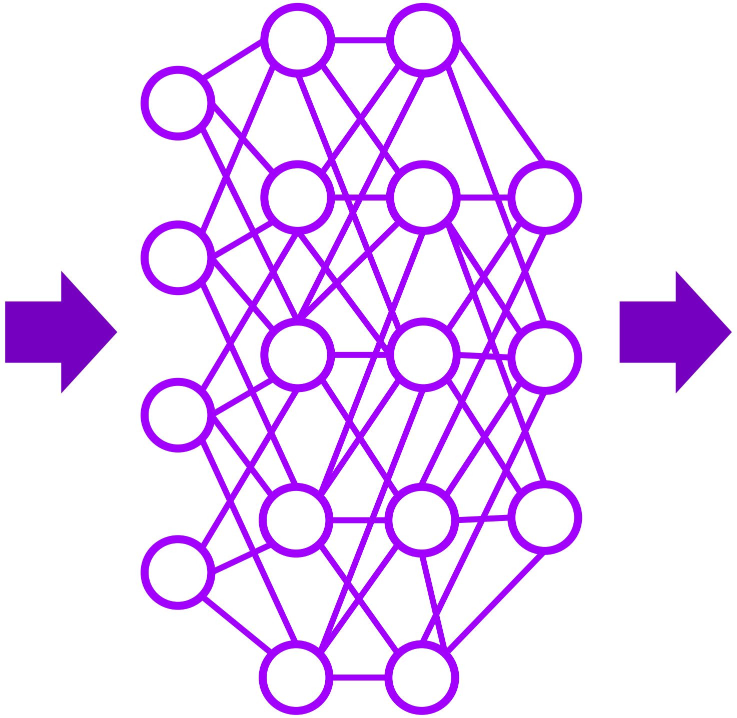
We can now build software that was previously impossible!

1. What do you see in the photo?



2. LLM

(ChatGPT 4)



3. LLM Response

The photo shows a dog with curly fur wearing a pair of headphones.

The dog appears to be enjoying the music, with its eyes closed and an expression of contentment.

The sky in the background is clear and blue, giving the image a cheerful and serene atmosphere.

The dog is also wearing a blue bandana around its neck.

LLMs vs. Java apps

LLMs

Data (text, photos, etc.) → LLMs (billion params)

- GPT-3: 175 billion params x 2bytes = 350GB
- Nobody fully understands how LLMs work!

Java apps

Java code → Bytecode → Machine code

- We understand how Java programs work ☺

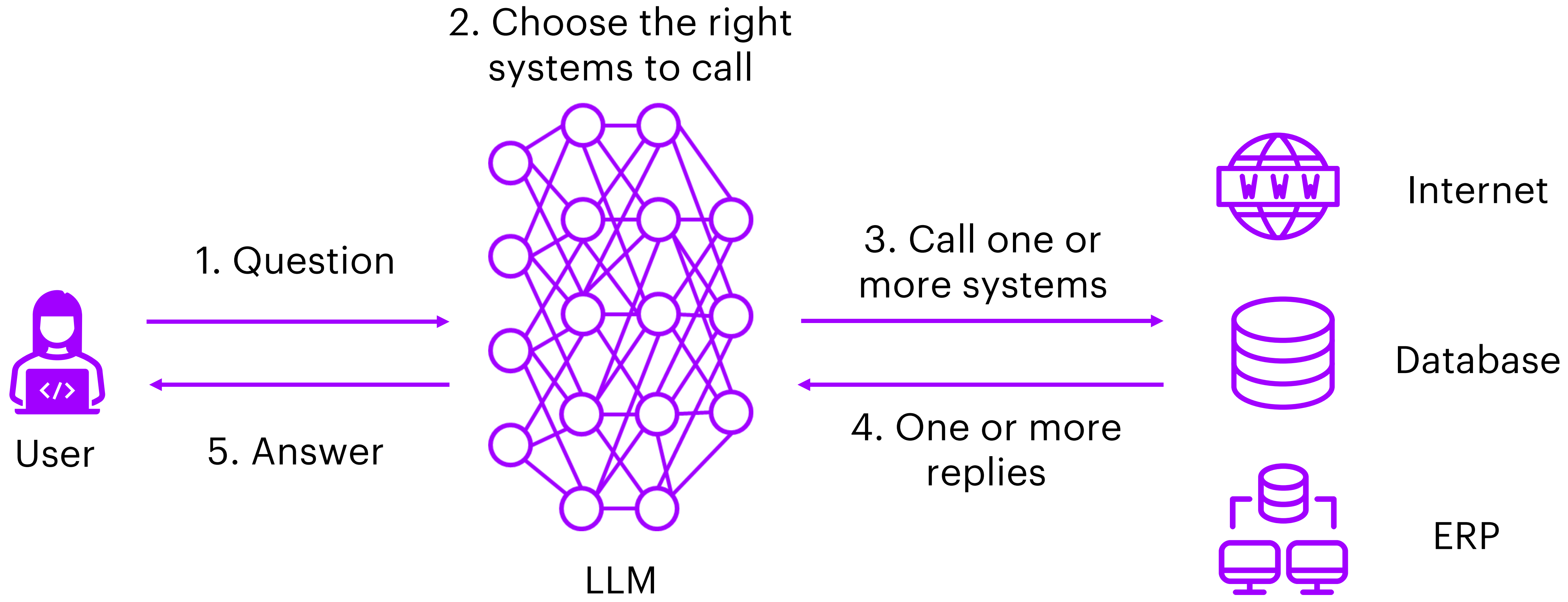
Coding Challenge!

Build a software application to:

“Find the cheapest flight tickets from Vilnius, Lithuania, to travel to the upcoming Oracle JavaOne conference”

Demo #5

Codeless: A new way of building software



Summarizing

What we learned for AI-Assisted Coding

Weaknesses

1. Generating new code that is functionally incorrect
2. Refactoring existing code will likely break the code
3. They are negatively impacting code quality

Strengths

1. Exploring new technologies & developing prototypes
2. Understanding code & enhancing its performance
3. Generating unit tests

Summarizing

Software Development Paradigms

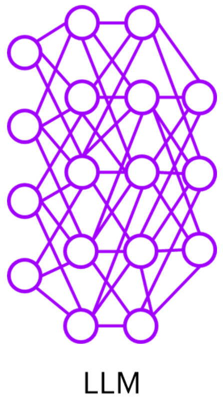
1. Code

Generated by humans and AI-Coding Assistants

```
public static void main(String[] args) {
    System.out.println(“Hello World!”);
}
```

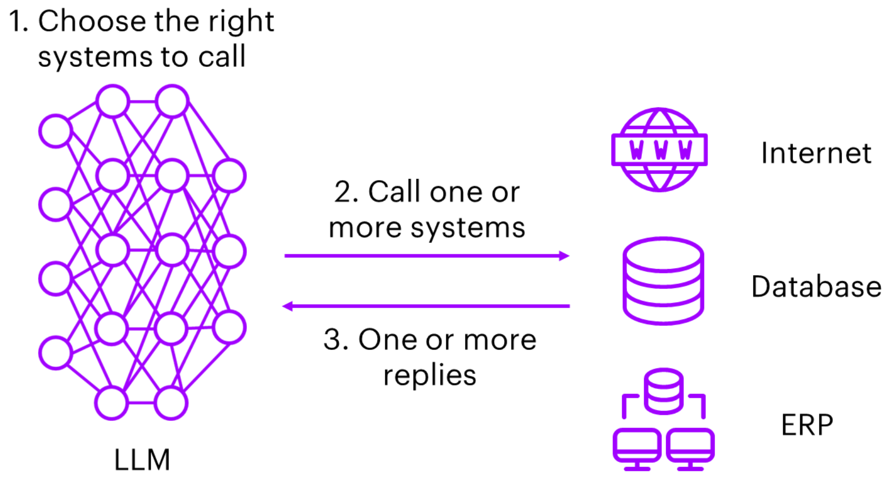
2. LLMs

Trained from vast data volumes



3. LLMs + Code

Blending the two development paradigms



1. Choose the right systems to call

2. Call one or more systems

3. One or more replies

LLM

Internet

Database

ERP

Will AI replace developers?

Image created using Midjourney

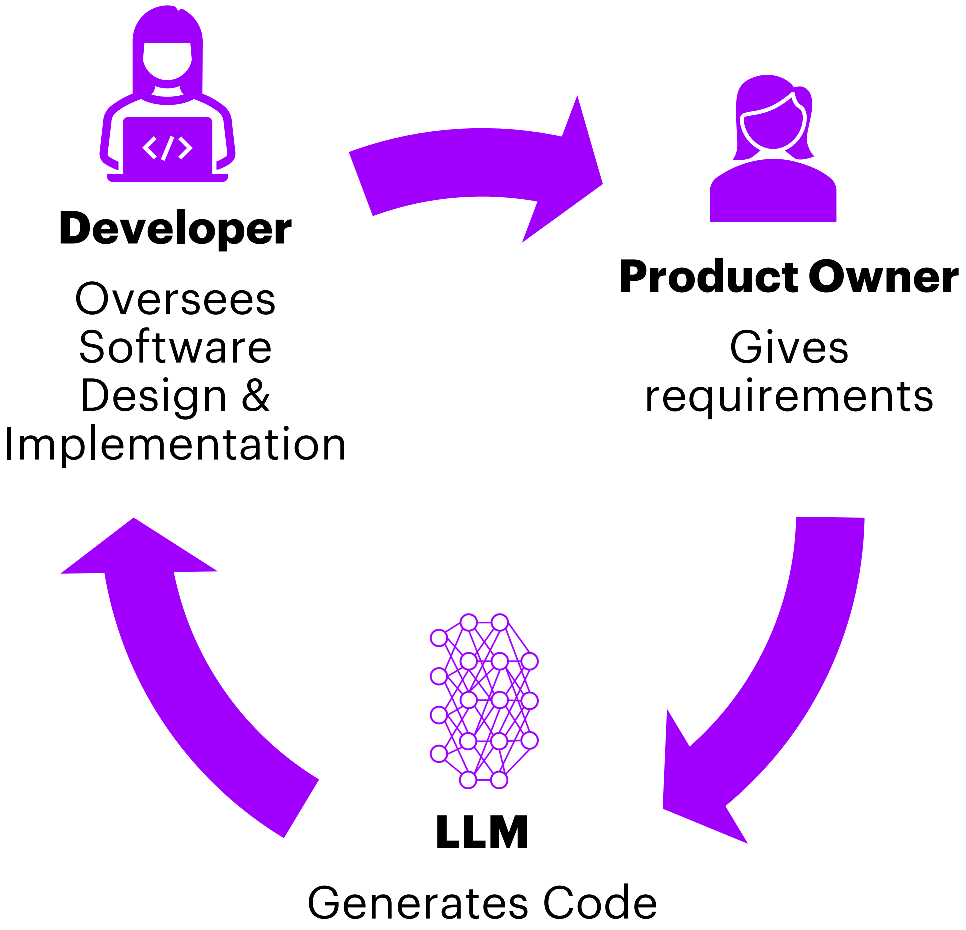
Human in the Loop

- 1912: First autopilot
- Today: Pilots focus on broader aspects of operations
  - Monitor trajectory, the weather, and the systems on board

Will AI replace developers?

Developers: “Human in the loop”

- Focusing on the bigger picture
- Mostly architecting,
  Less coding



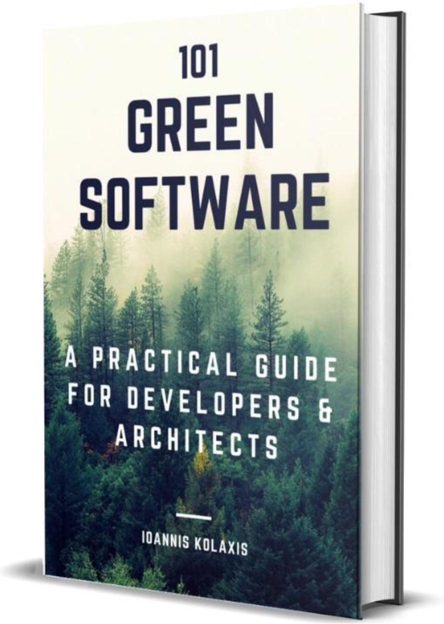
Become a 10x developer by harnessing Gen AI

1. Don’t forget unit tests and code reviews; AI-assisted coding tools are error-prone!
2. Use AI-assistants to understand code
3. Explore the new software development paradigms (LLMs + Code)

Who am I?

Ioannis Kolaxis - Director at Accenture

- Developer
  - From programming in BASIC with an Amstrad CPC 6128
  - To first job building Java Applets (in Java 1.2)
- Inventor: [5 Patents]
- Book author: Green Software



Visit website: kolaxis.dev