

Good Practices in ML 🙌

ELLIS Doctoral Symposium, Alicante 2022

Sebastian Dziadzio

 @sbdzdz

What will we focus on?

 Code smells

 Programming practices

 Tools

Code smells

Code smells

Duplicate code

Dead code

Large classes

Long signatures

Long methods

```
bool are_equal(int x, int xx)
{
    if (x>xx)
        return false;

    if(x<xx)
        return false;

    if(x==xx && xx==x)
        return true;

    return false
}
```



Code smells

Excessive comments

Mysterious names

Shotgun surgery

Magical programming

Excessive global variables



Programming Practices

Principles

KISS: Keep it simple, silly

YAGNI: You aren't gonna need it

DRY: Don't repeat yourself

“It seems that perfection is reached not when there is nothing left to add, but when there is nothing left to take away.”

Antoine de Saint Exupéry

Readability

Be suspicious of comments

Use descriptive names

Maintain documentation

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

Martin Fowler

Layout

Build modular, reusable components

Write modules like a newspaper

Start simple and iterate!

“Always code as if the person who ends up maintaining your code will be a violent psychopath who knows where you live.”

John Woods

Refactoring

Simplify, Simplify, Simplify

Improve maintainability

Pay up your technical debt!

“Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are not smart enough to debug it.”

Brian Kernighan

Refactoring

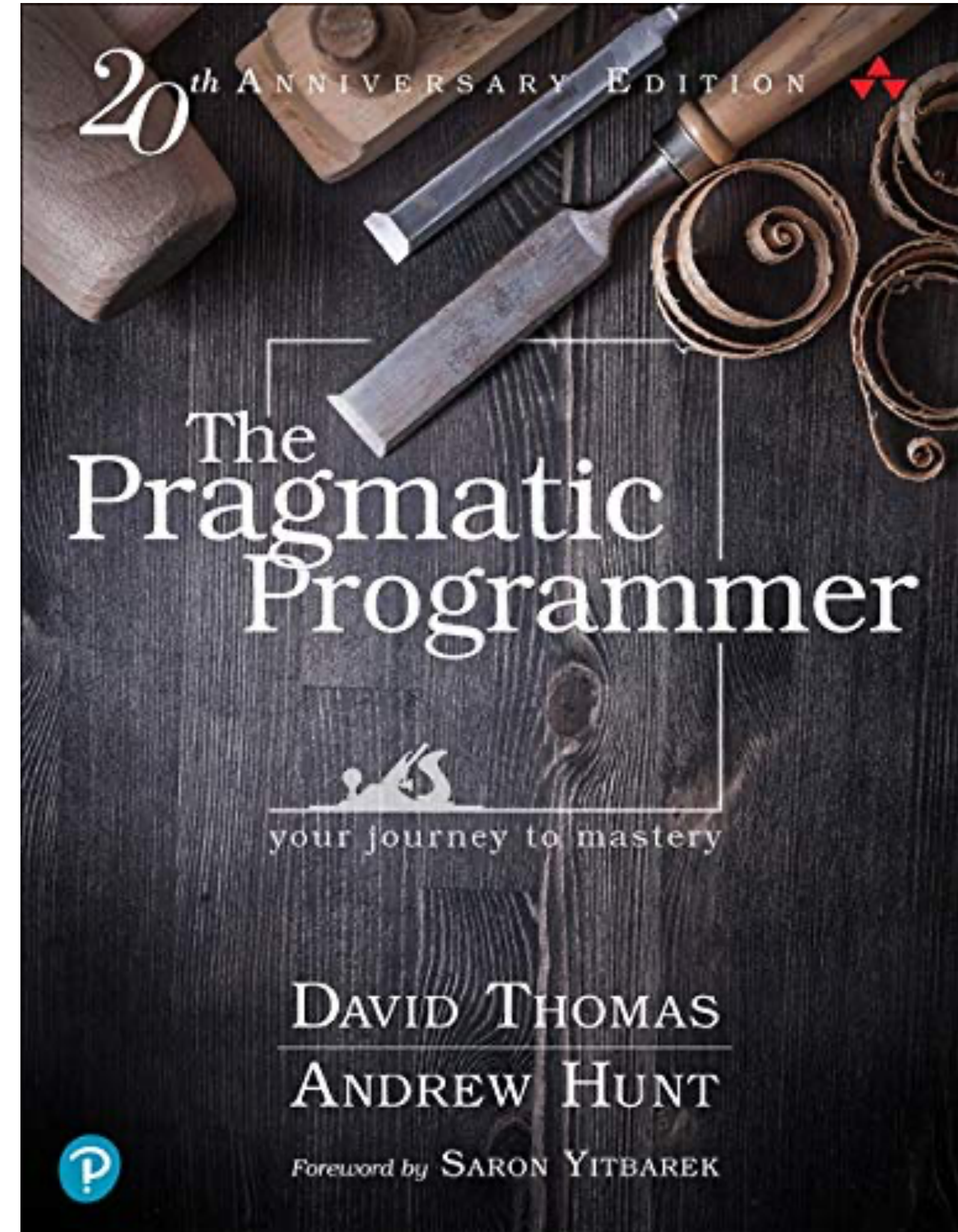
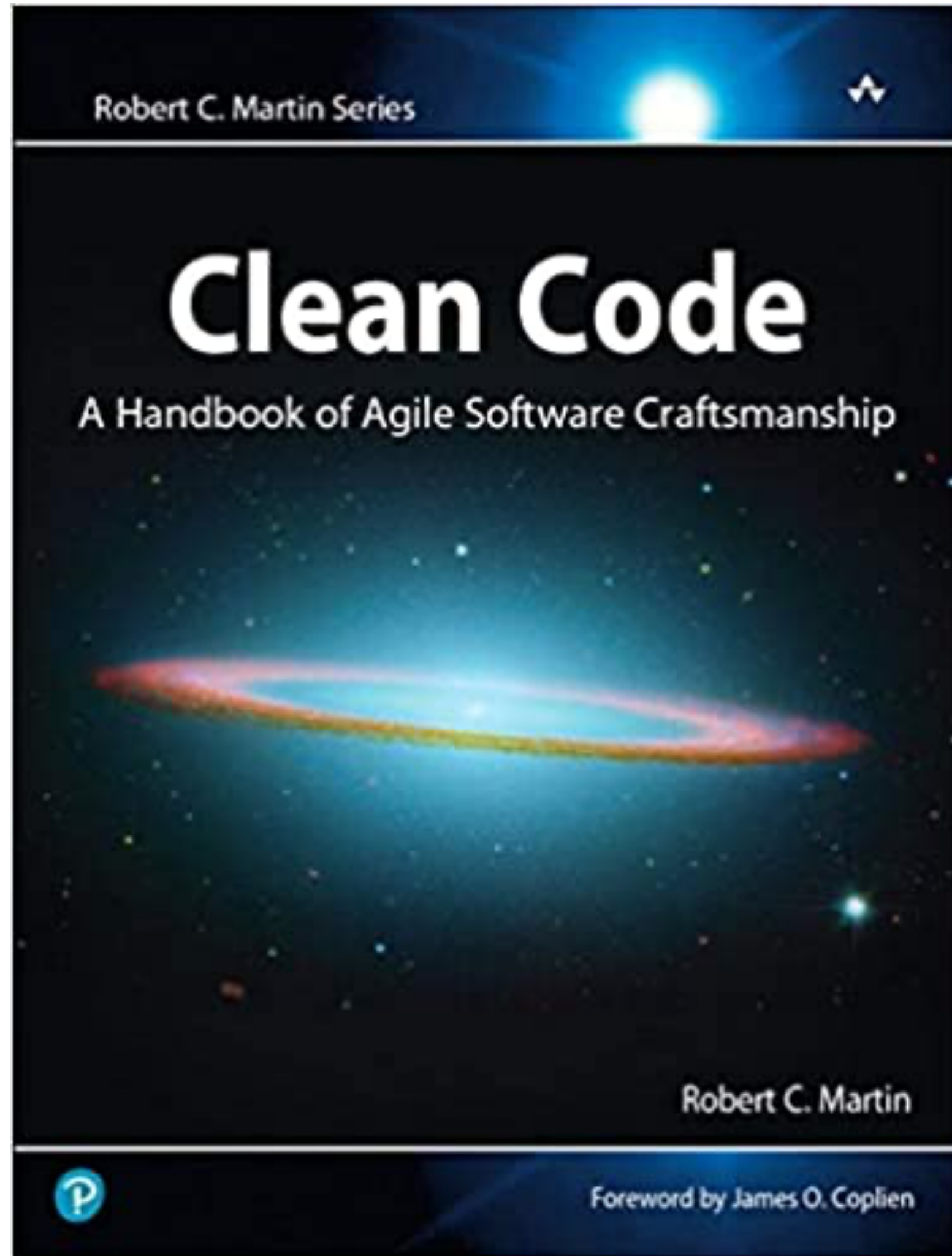
~~Simplify~~, ~~Simplify~~, Simplify

Improve maintainability

Pay up your technical debt!

“Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are not smart enough to debug it.”

Brian Kernighan



Tools

VS Code Python extension

Autocomplete and type analysis

Linting and formatting

Refactoring support

Import sorting

Debugger

Set conditional breakpoints

Track variables and expressions

Create launch configs

Control execution line by line

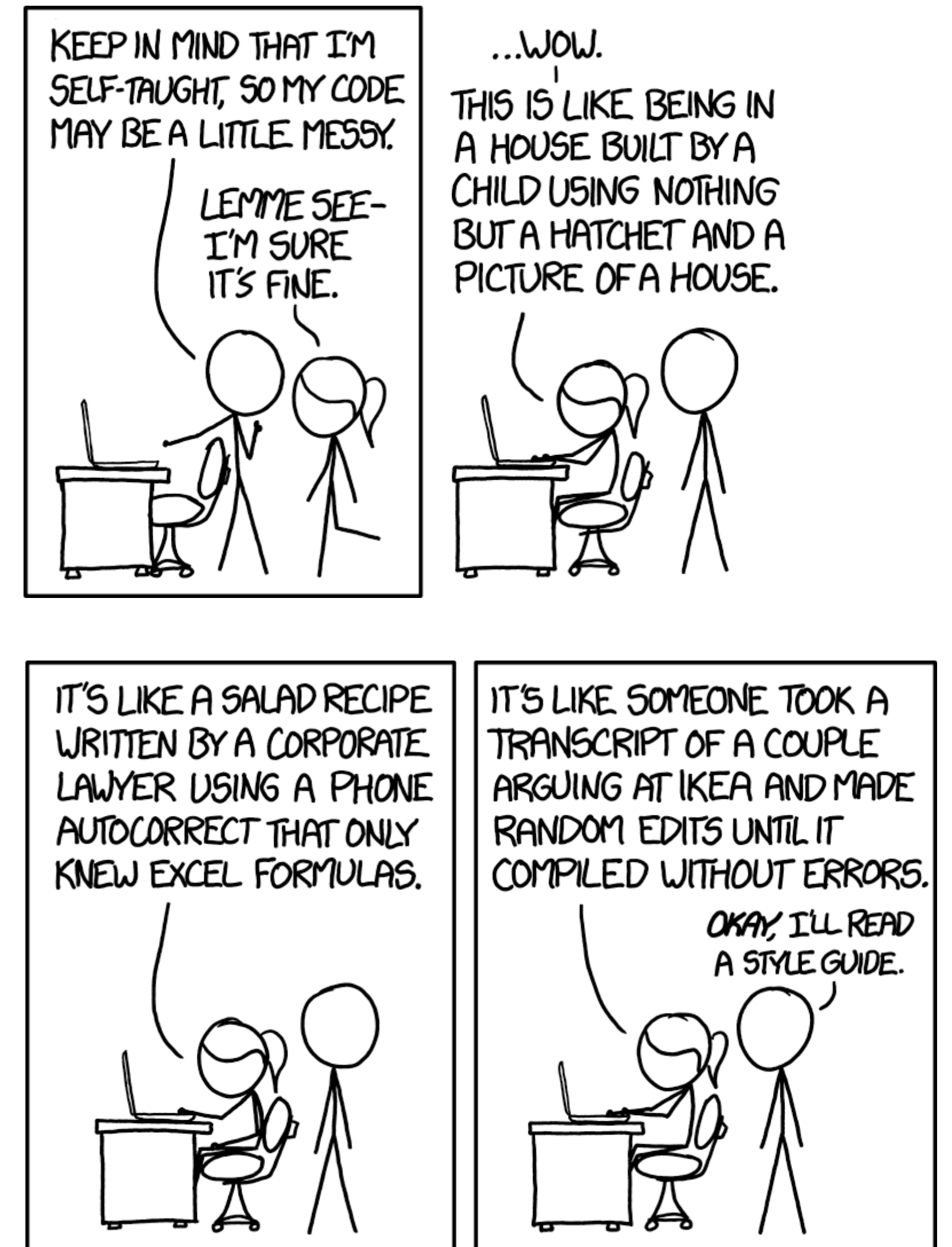
Linters

Flags syntax and style errors

Detects simple bugs and code smells

Can be whiny, feel free to adjust the rules

Choices: pylint, flake8



Formatter

Ensure a consistent style

Opinionated by design

Choices: black, yapf, autopep8



TABS VS. SPACES

Type checker

Pretend Python is statically typed

Make type annotations work for you

Choices: Mypy, Pytype, Pylint, Pyre

CI

Surprisingly easy to configure

Run tests, linters, formatters

Generate code coverage reports



GitHub Actions

Code review

Best way to improve

Learning to read code is essential

Takes some practice to do well!



I Am Developer

@iamdeveloper

10 lines of code = 10 issues.

500 lines of code = "looks fine."

Code reviews.

1:58 PM - 5 Nov 2013

8,339 Retweets 5,541 Likes



114

8.3K

5.5K

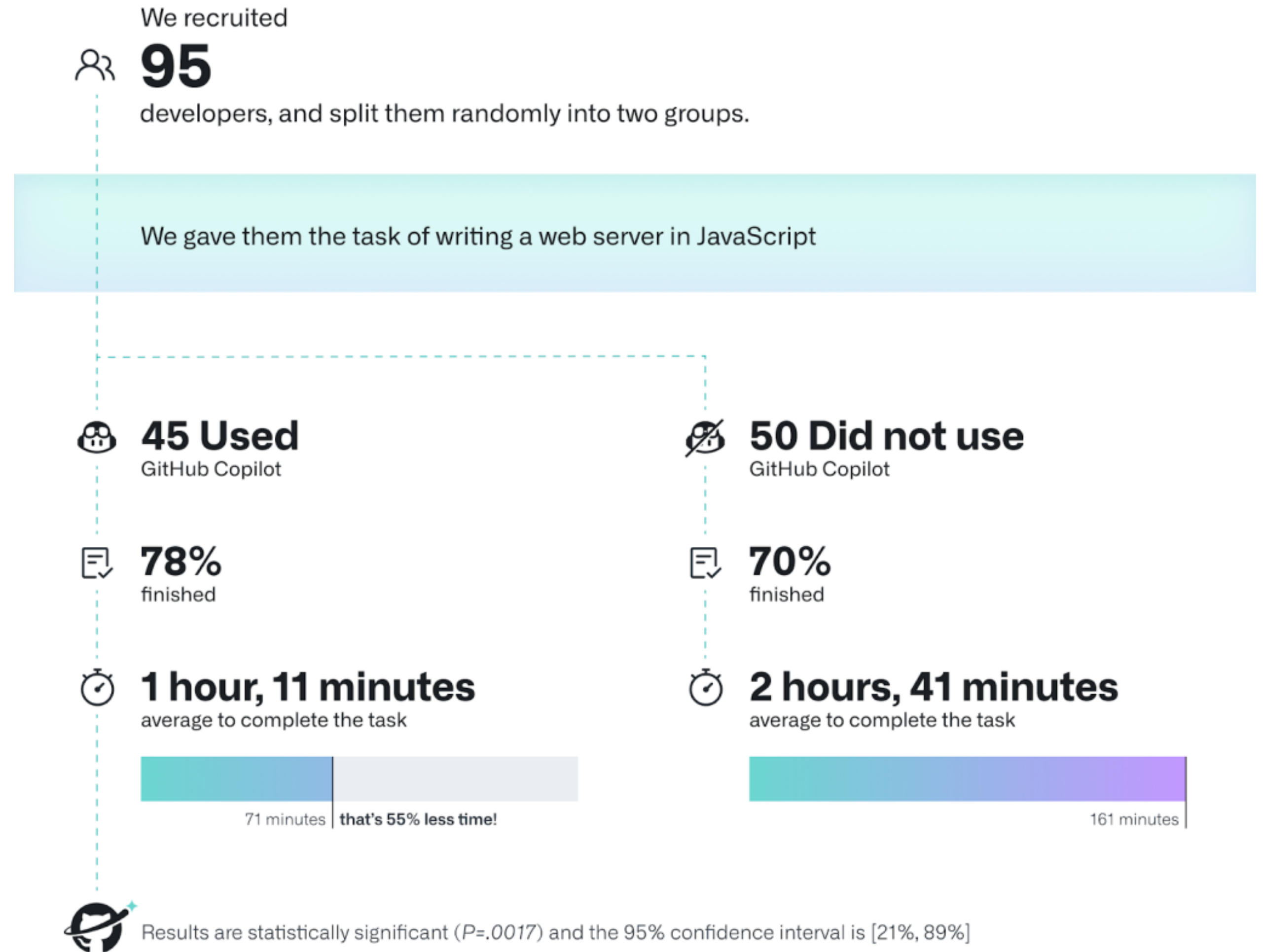


GitHub Copilot

ML based assistant

Helps with boilerplate

Free for students!



Pathlib

Object oriented paths

Clean syntax, OS agnostic

No more `os.path.join` 😊

```
from pathlib import Path

path = Path("~/Documents")
print(path)
# ~/Documents

path = path.expanduser()
print(path)
# /home/sebastian/Documents

print(path.is_dir())
# True

project_name = "mnist"
file_name = "train.py"
path = path / f"projects/{project_name}/{file_name}"
print(path)
# /Users/sebastian/Documents/projects/mnist/train.py

print(path.is_file())
# True

print(path.name, path.stem, path.suffix)
# train.py train .py
```

Thank you!

 @sbdzdz