Thank you to our Sponsors!

Developer!
DeveloperDeveloper!
UK Community Events

UNIVERSITY OF HULL

black marble

turbo360

{ } NDC Conferences

Sage

Sam

iO associates

Help! I've fallen and I can't get up

hey hi hello

derek graham
@deejaygraham.bsky.social
@deejaygraham@hachyderm.io


Sage

I'm doing a lot of stuff with computers right now

WeThinkCode_

Mentors Handbook 2022

A guide to the second year mentorship program.

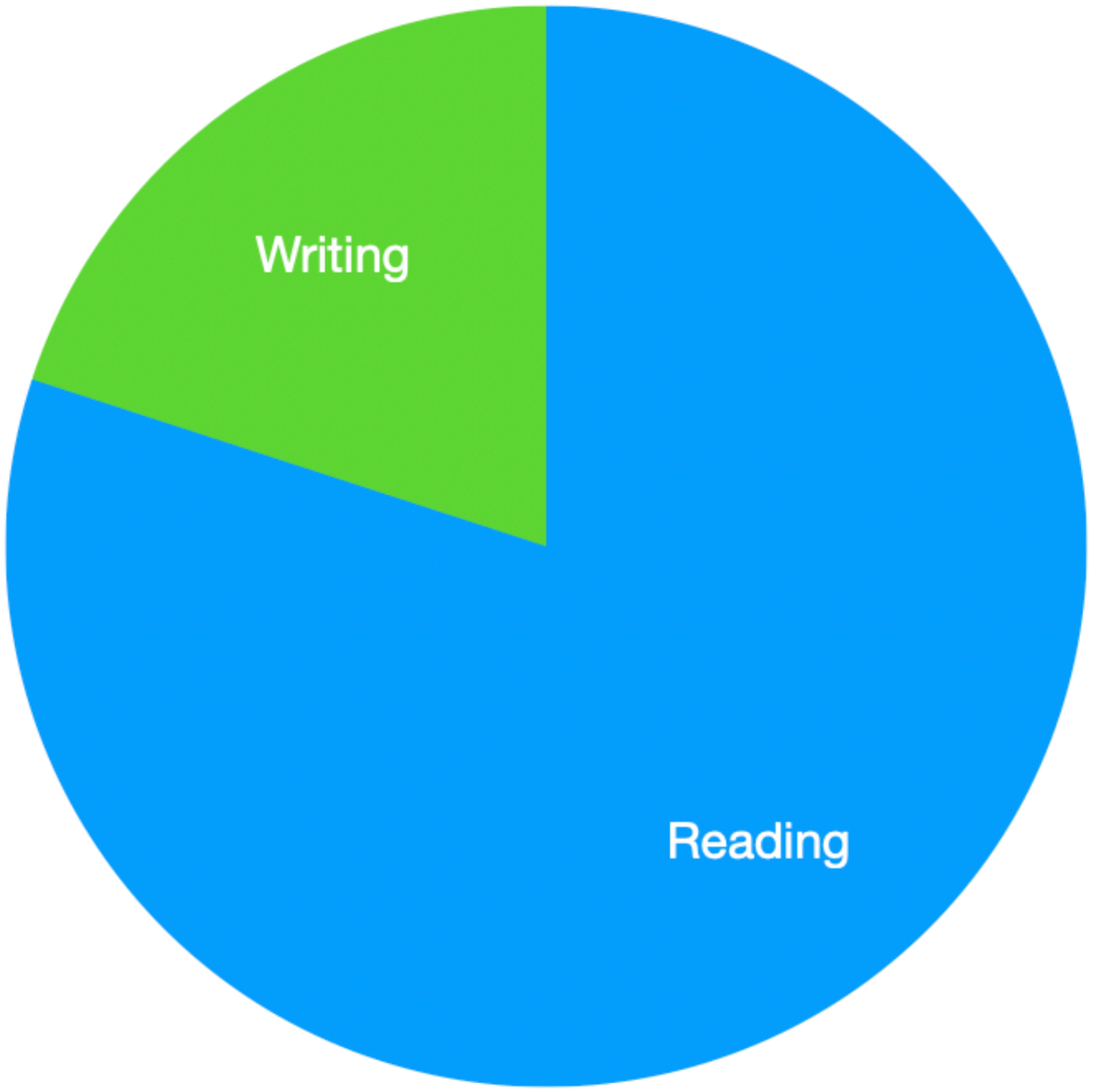

AQUA MARINA
FUSION
DS LIGHT TECH
SPORTS III

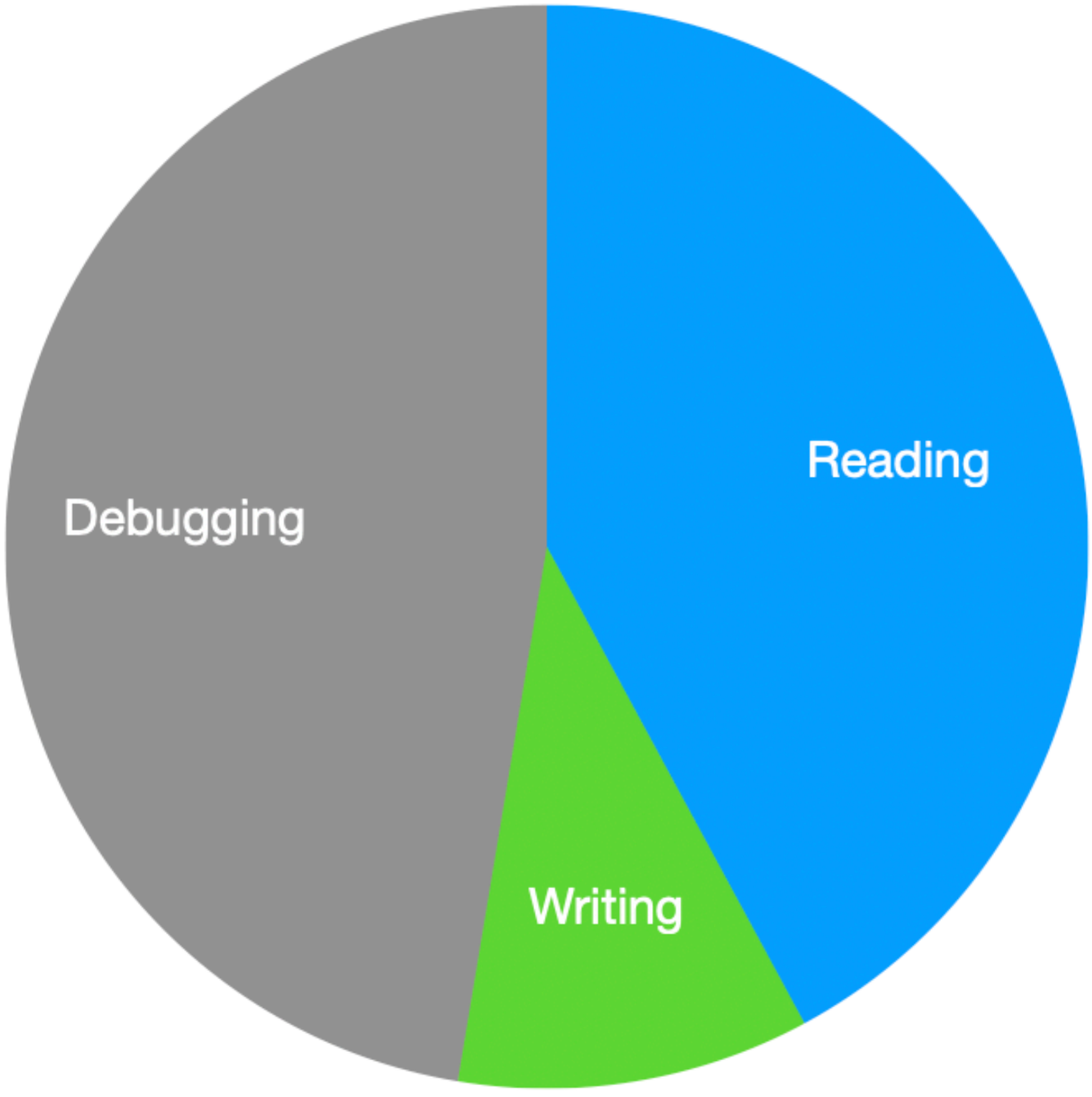
Stuck?

We are (nearly)
always wrong

“Always wanted to travel back in time to try fighting a younger version of yourself ? Software development is the career for you !”

@Loh


Writing
Reading


Reading
Writing
Debugging

CD-ROM Included

Debugging Applications

The Bugslayer's guide to finding and fixing coding errors in Microsoft® Windows®–based applications

John Robbins

“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, by definition, not smart enough to debug it.”

Brian Kernighan

Assumptions

CD-ROM Included

Debugging Applications is Hard

The Bugslayer's guide to finding and fixing coding errors in Microsoft® Windows®–based applications

John Robbins

(Un)Certainty

~~understanding~~

SPEED UP!!!!!!!!


SLOW
DOWN

Progress ...

...Investigation

Change !!!

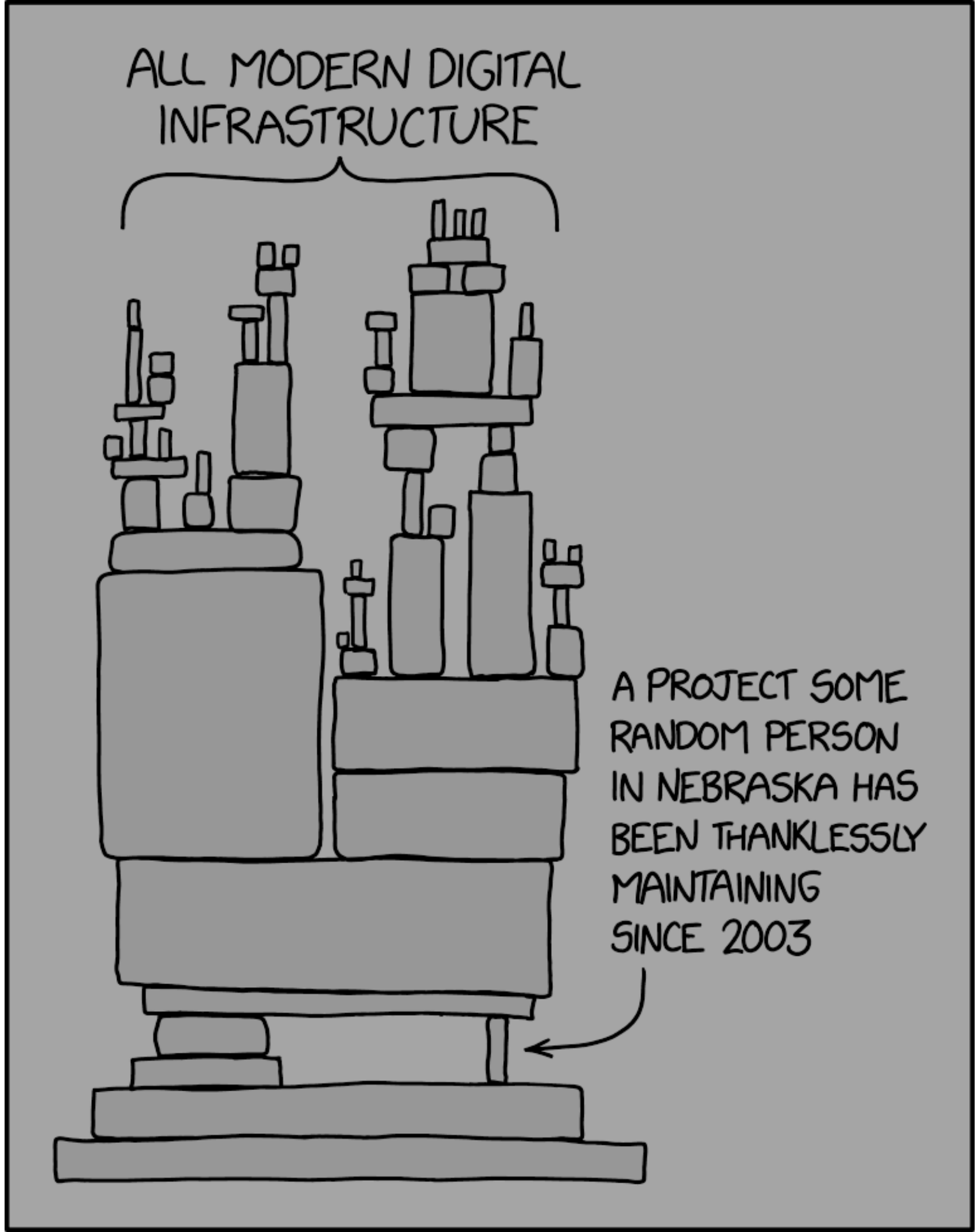
Change !!! ...

Something!!!

```
while(1):
    change();
```

Code or Understanding

Your Fault ?


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

Smaller Steps

Better

Feedback

Assumptions

Hard

ThisFileExits.txt

Make a List

(or two)

- Automated Tests
- Manual Tests
- Breakpoints
- Print(“I am here”)
- Log(“also here”)
- Perturbation

Bugs Hide in the Stuff We Don't Check

“The first principle is that you must not fool yourself - and you are the easiest person to fool”

Richard Feynman

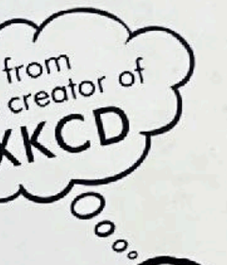
- Hypothesis
- Design Experiment
- Run Experiment
- Check outcome

- Is the table correct?
- Is the data sensible?
- Is this query running?
- What is the SQL?
- Can we run it manually?
- Is there a firewall?

THE UK AND US No.1 BESTSELLER

SERIOUS SCIENTIFIC ANSWERS TO ABSURD HYPOTHETICAL QUESTIONS

from the creator of XKCD



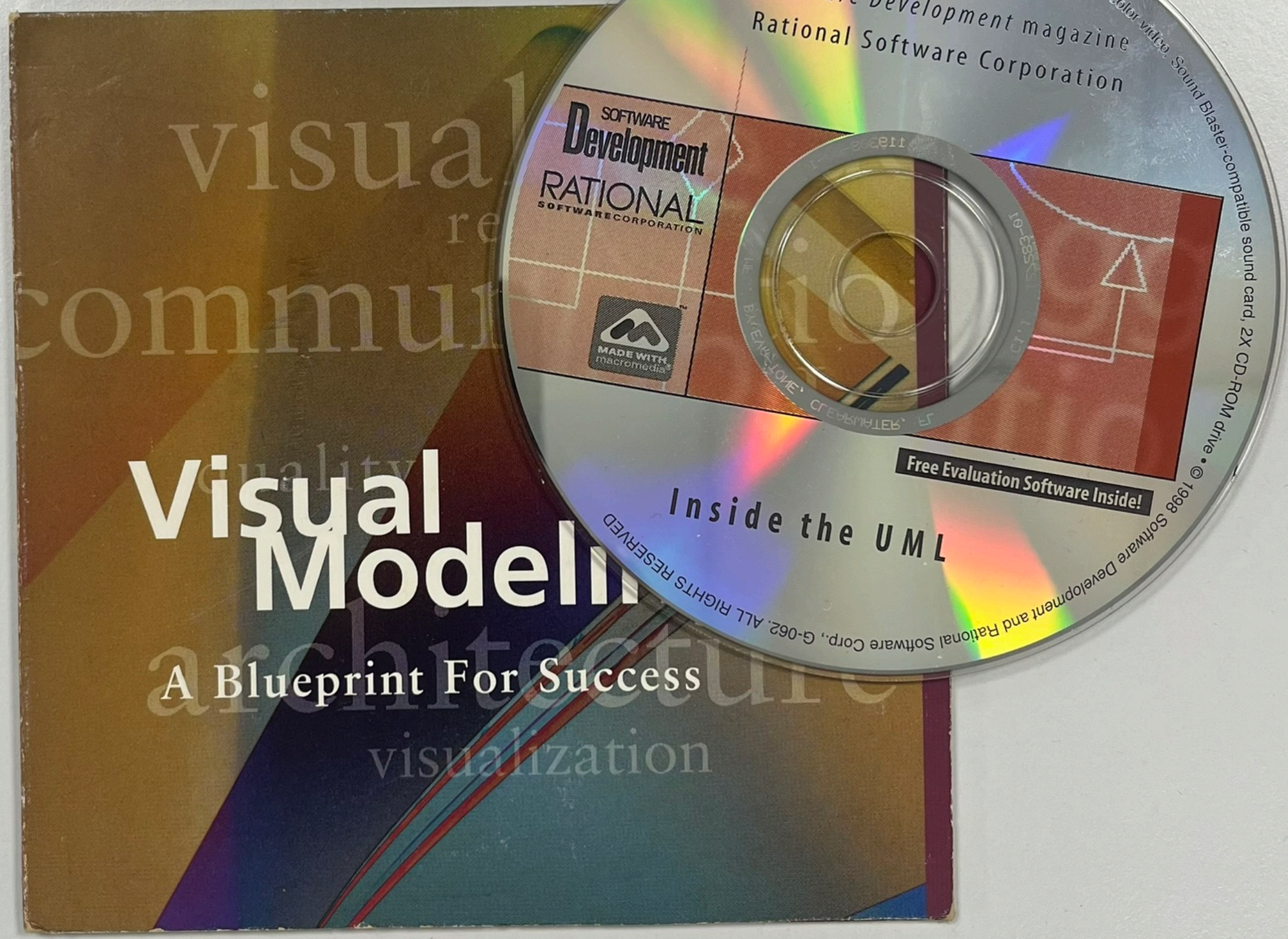
'NERD ROYALTY'
BEN GOLDACRE

'BRILLIANT'
ROLLING STONE

WHAT IF?

RANDALL MUNROE

Write Stuff Down


visual
communi
quality
Visual
Modelin
architecture
A Blueprint For Success
visualization
Minimum requirements: Windows 95/NT, 16MB RAM, 256-color video, Sound Blaster-compatible sound card, 2X CD-ROM drive • © 1998 Software Development and Rational Software Corp., G-062. ALL RIGHTS RESERVED
Software Development magazine
Rational Software Corporation
SOFTWARE
Development
RATIONAL
SOFTWARE CORPORATION
MADE WITH macromedia
Free Evaluation Software Inside!
Inside the UML

Get It Out of Your Head

Yaks & Ducks & Rabbits.

Yaks &

Ducks &

Rabbits.

YAK SHAVING DAY.

32-bit computing

From Wikipedia, the free encyclopedia

In computer architecture, **32-bit computing** refers to computer systems with a processor, memory, and other major system components that operate on data in 32-bit units.[1][2] Compared to smaller bit widths, 32-bit computers can perform large calculations more efficiently and process more data per clock cycle. Typical 32-bit personal computers also have a 32-bit address bus, permitting up to 4 GB of RAM to be accessed, far more than previous generations of system architecture allowed.[3]

32-bit designs have been used since the earliest days of electronic computing, in experimental systems and then in large mainframe and minicomputer systems. The first hybrid 16/32-bit microprocessor, the Motorola 68000, was introduced in the late 1970s and used in systems such as the original Apple Macintosh. Fully 32-bit microprocessors such as the HP FOCUS, Motorola 68020 and Intel 80386 were launched in the early to mid 1980s and became dominant by the early 1990s. This generation of personal computers coincided with and enabled the first mass-adoption of the World Wide Web. While 32-bit architectures are still widely-used in specific applications, the PC and server market has moved on to 64 bits with x86-64 and oth 64-bit architectures since the mid-2000s with installed memory often exceeding the 32-bit 4G RAM address limits on entry level computers. The latest generation of smartphones have also switched to 64 bits.

Computer architecture bit widt

Bit

1 · 4 · 8 · 12 · 16 · 18 · 24 · 26 · 30 · 31 · **32**
· 45 · 48 · 60 · 64 · 128 · 256 · 512 · bit sli

Application

8 · 16 · 32 · 64

Binary floating-point precision

16 (×½) · 24 · 32 (×1) · 40 · 64 (×2) · 80 ·
(×4) · 256 (×8)

Decimal floating-point precision

32 · 64 · 128

Range for storing integers [edit]

A 32-bit register can store 2^{32} different values. The range of integer values that can be stored in 32 bits depends on the integer

Illiac Suite

From Wikipedia, the free encyclopedia

Illiac Suite (later retitled **String Quartet No. 4**)[1] is a 1957 composition for string quartet which is generally agreed to be the fi score composed by an electronic computer.[2] Lejaren Hiller, in collaboration with Leonard Isaacson, programmed the ILLIAC I computer at the University of Illinois at Urbana–Champaign (where both composers were professors) to generate compositiona material for his String Quartet No. 4.

The piece consists of four movements, corresponding to four experiments: the first is about the generation of *cantus firmi*, the s generates four-voice segments with various rules, the third deals with rhythm, dynamics and playing instructions, and the fourt various models and probabilities for generative grammars or Markov chains (see stochastic music).[3]

References [edit]

1. ^ Andrew Stiller, "Hiller, Lejaren (Arthur)", *Grove Music Online* (reviewed December 3, 2010; accessed December 14, 2014).
2. ^ Denis L. Baggi, "The Role of Computer Technology in Music and Musicology Archived 2011-07-22 at the Wayback Machine", *lim.dico.unimi.it* (December 9, 1998).
3. ^ Lejaren A. Hiller and Leonard M. Isaacson, *Experimental Music: Composition With an Electronic Computer*, second edition (New McGraw-Hill, 1959): 5–7. Reprinted, Westport, Conn.: Greenwood Press, 1979. ISBN 978-0-313-22158-3.

External links [edit]

- "Lejaren Hiller – Illiac Suite for String Quartet [1/4]" on YouTube December 4, 2011. See also: 2/4, 3/4, and 4/4.
- Sandred, Örjan; Laurson, Mikael; Kuuskankare, Mika. "Revisiting the Illiac Suite – a rule based approach to stochastic processes" (PDF).

Isolate

- What has been added or changed?
- What are you least confident about?
- Where is there “magic” ?
- Where are the most assumptions?
- What can’t possibly be wrong?

Binary Search

NAME

git-bisect - Use binary search to find the commit that introduced a bug

SYNOPSIS

```
git bisect <subcommand> <options>
```

DESCRIPTION

The command takes various subcommands, and different options depending on the subcommand:

```
git bisect start [--term-(bad|new)=<term-new> --term-(good|old)=<term-old>]
        [--no-checkout] [--first-parent] [<bad> [<good>...]] [--] [<pathsp
git bisect (bad|new|<term-new>) [<rev>]
git bisect (good|old|<term-old>) [<rev>...]
git bisect terms [--term-(good|old) | --term-(bad|new)]
git bisect skip [(<rev>|<range>)...]
git bisect reset [<commit>]
git bisect (visualize|view)
git bisect replay <logfile>
git bisect log
git bisect run <cmd> [<arg>...]
git bisect help
```

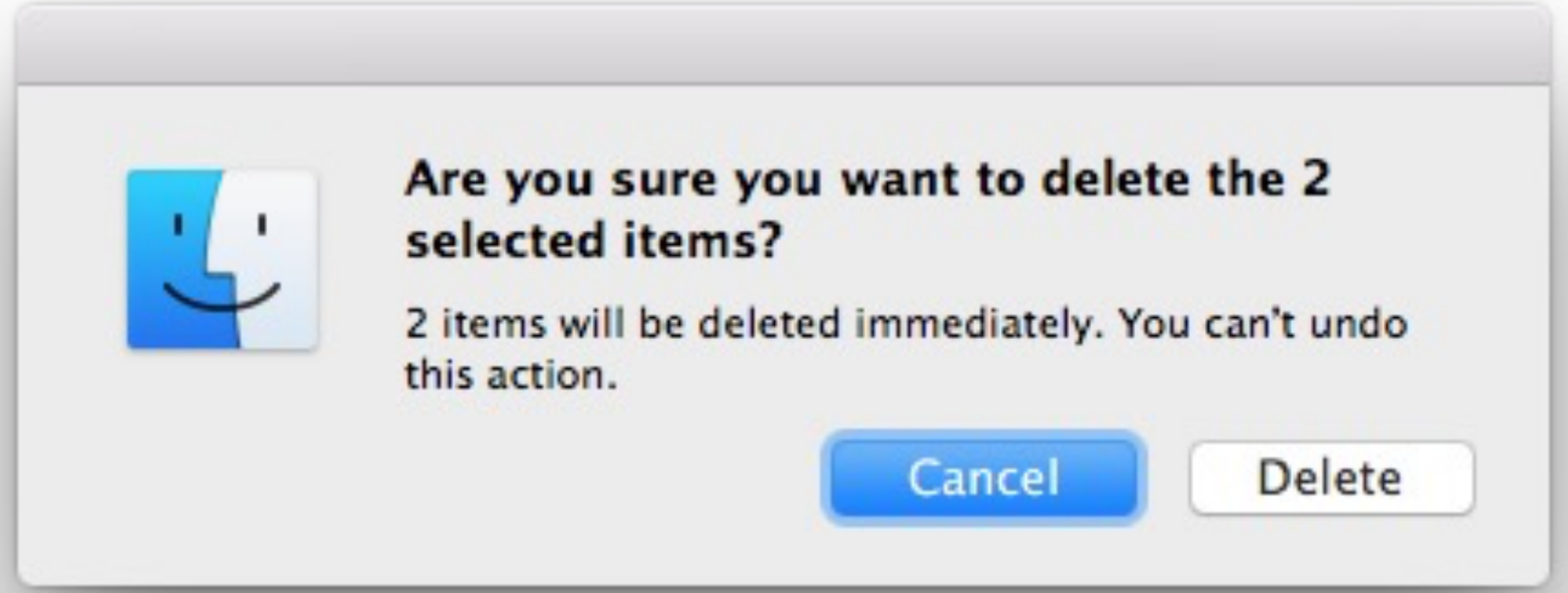
This command uses a binary search algorithm to find which commit in your project's history introduced a bug. You use it by first telling it a "bad" commit that is known to contain the bug, and a "good" commit that is known to be before the bug was introduced. Then `git bisect` picks a commit between those two endpoints and asks you whether the selected commit is "good" or "bad". It continues narrowing down

Make it Smaller

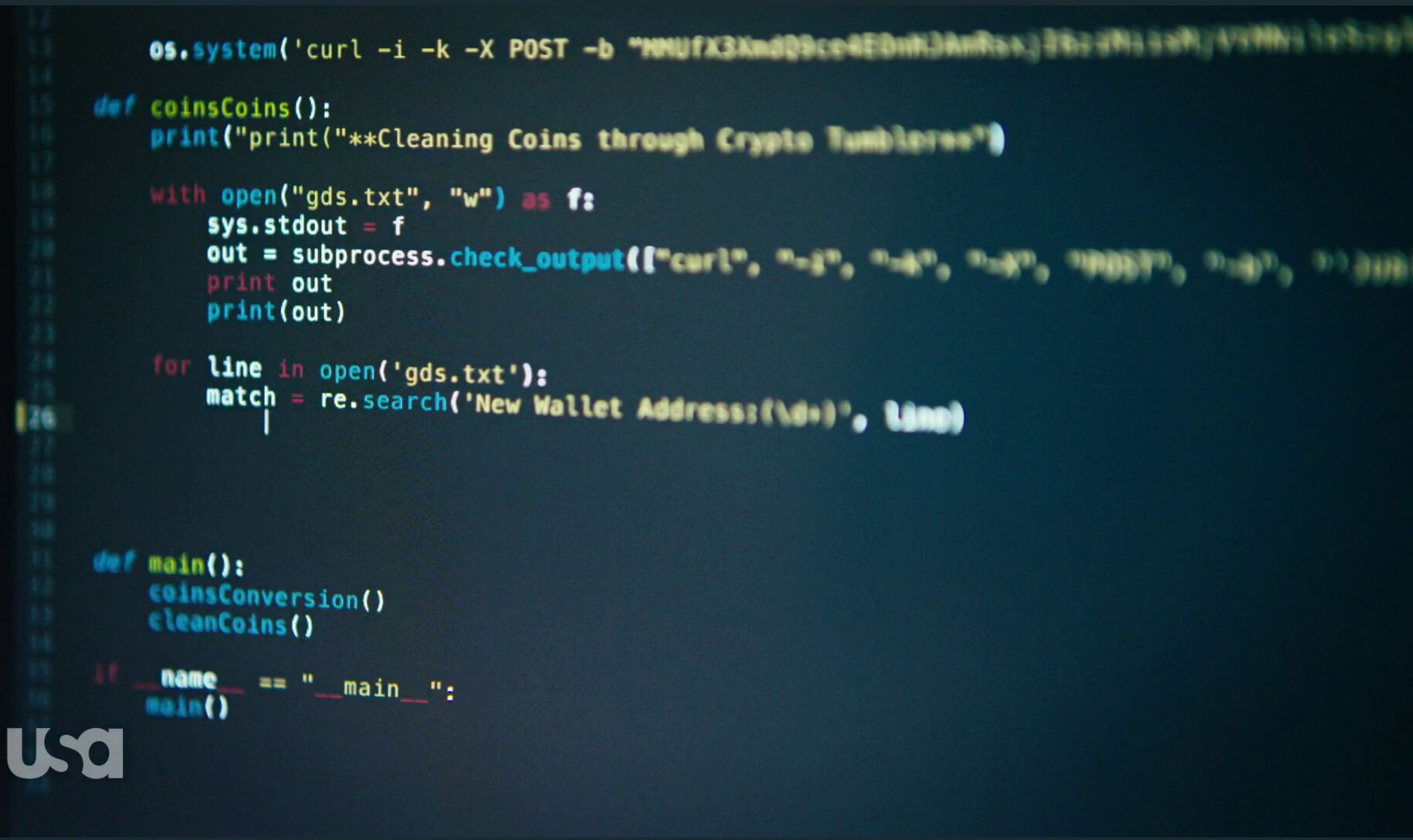
Make it

Smaller

Make it Smaller


Are you sure you want to delete the 2 selected items?
2 items will be deleted immediately. You can't undo this action.
Cancel
Delete

```
True:

    if button_a.was_pressed():
        play = not play

    if play:
        note_name = random.choice(notes)

        if random.randint(0, 100) <= change_octave_pc:
            octave = random.choice(octaves)

        if random.randint(0, 100) <= change_duration_pc:
            duration = random.choice(durations)

        visualize = note_name != 'R'
        if visualize:
            x = random.randint(0, 4)
            y = 5 - octave

            display.set_pixel(x, y, 9)

        note = format_note(note_name, octave, duration)
        music.play(note, wait=True)

        if visualize:
            display.set_pixel(x, y, 0)
```


os.system('curl -i -k -X POST -b "
def coinsCoins():
print("print("**Cleaning Coins through Crypto Tumblers**")
with open("gds.txt", "w") as f:
sys.stdout = f
out = subprocess.check_output(["curl", "-i", "-k", "-X", "POST", "-b",
print out
print(out)
for line in open('gds.txt'):
match = re.search('New Wallet Address:(\d+)', line)
def main():
coinsConversion()
cleanCoins()
if __name__ == "__main__":
main()

Write a Test

Write many Tests

Embed Tests

```
if (thingExists) {
  doImportantThing(thingExists);
}
```

```
if (thingExists) {
  doImportantThing(thingExists);
} else {
  Log(“Thing doesn’t exist!!”);
}
```

Start

...somewhere

Spike

$@?&%! First Draft

“Almost all good writing begins with terrible first efforts”

Annie Lamott

Pretend

Photographer: Iain Macmillan / © Yoko Ono

Distract

TIREDNESS
CAN KILL
TAKE A
BREAK

'The darker the hour, the better
imagined utopian fiction. A wonderful
we've come to expect from Cory Doctorow and
WILLIAM GIBSON

walkaway

A NOVEL

cory doctorow

- Go Home
- Nap
- Exercise
- Read a Book
- Different (Mechanical) Problem

“Almost everything will work again if you unplug it for a few minutes, including you.”

Annie Lamott

“Ideas **churn** around below the threshold of **consciousness**… it is during this time that **unusual** connections are likely to be made”

Mihaly Csikszentmihalyi

Beware!!

Get Ready for Next Time

Learn...

- New ways of solving problems
- Pair with someone new
- Read someone else's code
- A new language?

But...!?!

...won't AI

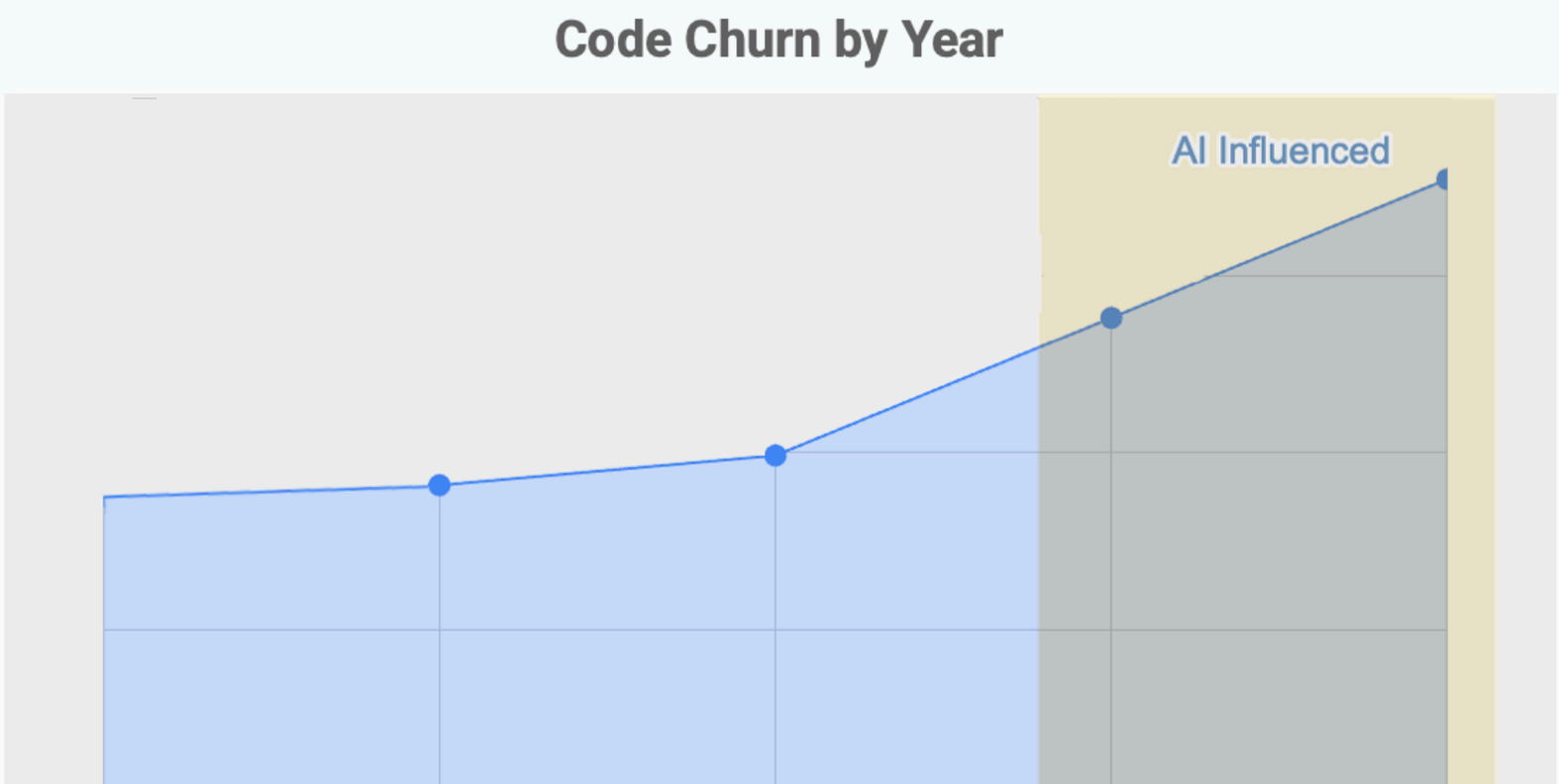
fix it?

Coding on Copilot: 2023 Data Suggests Downward Pressure on Code Quality

Including 2024 projections for specific code reuse

GitHub and other sources have reported more than 50% of developers adopting AI Assisted-development during 2023. What these sources haven't reported is how the composition of code changes when AI is used.

We examine 4 years worth of data, encompassing more than 150m changed lines of code, to determine how AI Assistants influence the quality of code being written. We find a significant uptick in churn code, and a concerning decrease in code reuse.



Recommends

The Pragmatic Programmer

from journeyman
to master

Andrew Hunt
David Thomas

Foreword by Ward Cunningham

NEW YORK TIMES BESTSELLER

UPDATED AND EXPANDED

"Words leap off the page."
— *USA Today*

brain rules

12 Principles for Surviving and Thriving
at Work, Home, and School

JOHN MEDINA

Includes link to *Brain Rules* film

The Pragmatic Programmers

Pragmatic Thinking & Learning

Refactor Your Wetware

Andy Hunt

Questions

Answers

That's all Folks!