CUPID – for joyful coding

Daniel Terhorst-North

@tastapod

motivation: not so SOLID

1. PubConf
   - poking the bear
2. Extreme Tuesday Club
   - accepting a challenge
3. Online conferences
   - working to a deadline!

setting the scene

who are we writing code for?

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

– Martin Fowler

who are we writing code for?

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

– Martin Fowler

Can we do better than understand?

“Habitability is the characteristic of source code that enables [people] to understand its construction and intentions and to change it comfortably and confidently.

“Habitability makes a place liveable, like home.”

– Richard P. Gabriel

habitable sounds better! What about joyful?

Think about a codebase that is joyful to work with

Can we describe what makes it joyful?

What kind of properties does joyful code have?

properties over principles

Principles are rules or guidelines

- They define conditions or boundaries
- Your code either conforms to the conditions or it is wrong

Properties are qualities or characteristics

- They define a goal or centre to move towards
- Your code is only ever closer to or further from the centre

properties for properties

practical

- easy to articulate, easy to assess, easy to adopt

human

- from the perspective of readers as well as writers

layered

- guidance for beginners, nuance for the more experienced

CUPID – for joyful coding

Composable	– plays well with others
Unix philosophy	– does one thing well
Predictable	– does what you expect
Idiomatic	– feels natural
Domain-based	– in language and structure

Composable

Code that is easy to use gets used, and used, and used!

Small “surface area”

- less to learn, less to go wrong, less to conflict

Intention-revealing name and purpose

- easy to discover, easy to evaluate

Minimal dependencies

- “You wanted a banana but you got a whole gorilla” – Joe Armstrong

Unix philosophy

"Make each program do one thing well" – and only one thing

- `ls` lists file details (but it doesn't inspect files!)

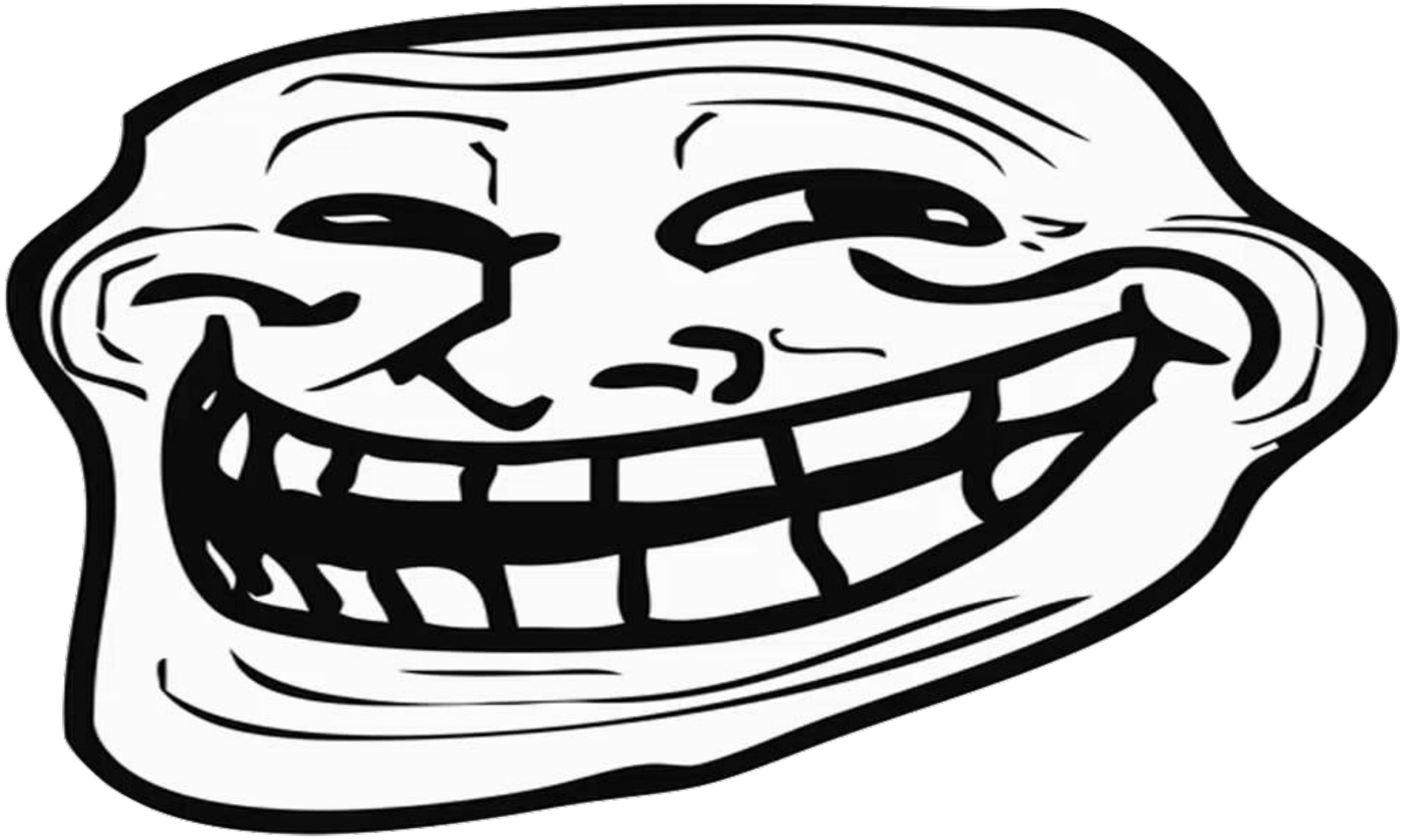
Together with composability, there is nothing you can't do!

- *"Expect the output of every program to become the input to another"*
- `cat | grep | sed | sort | uniq | …`

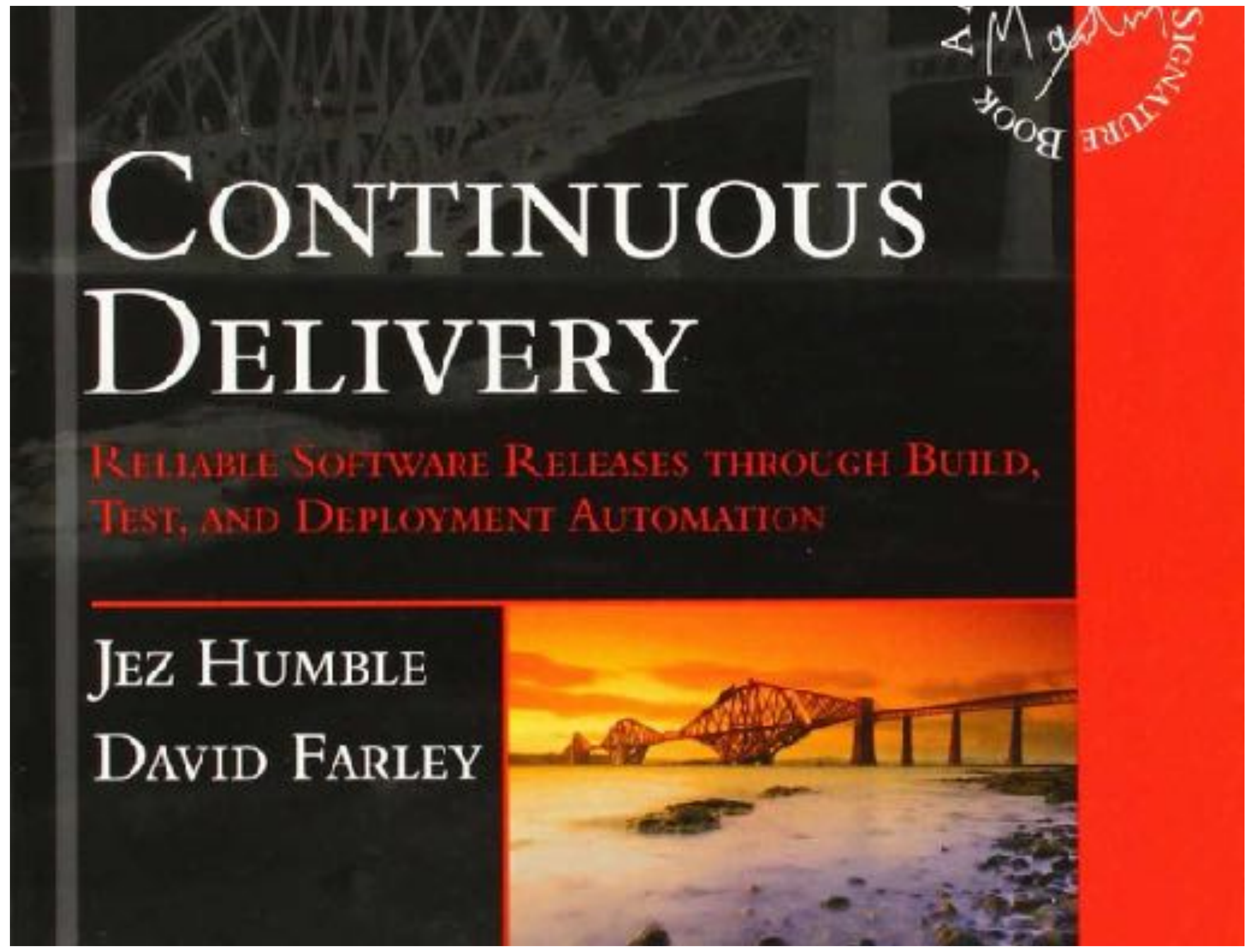
Different from SRP

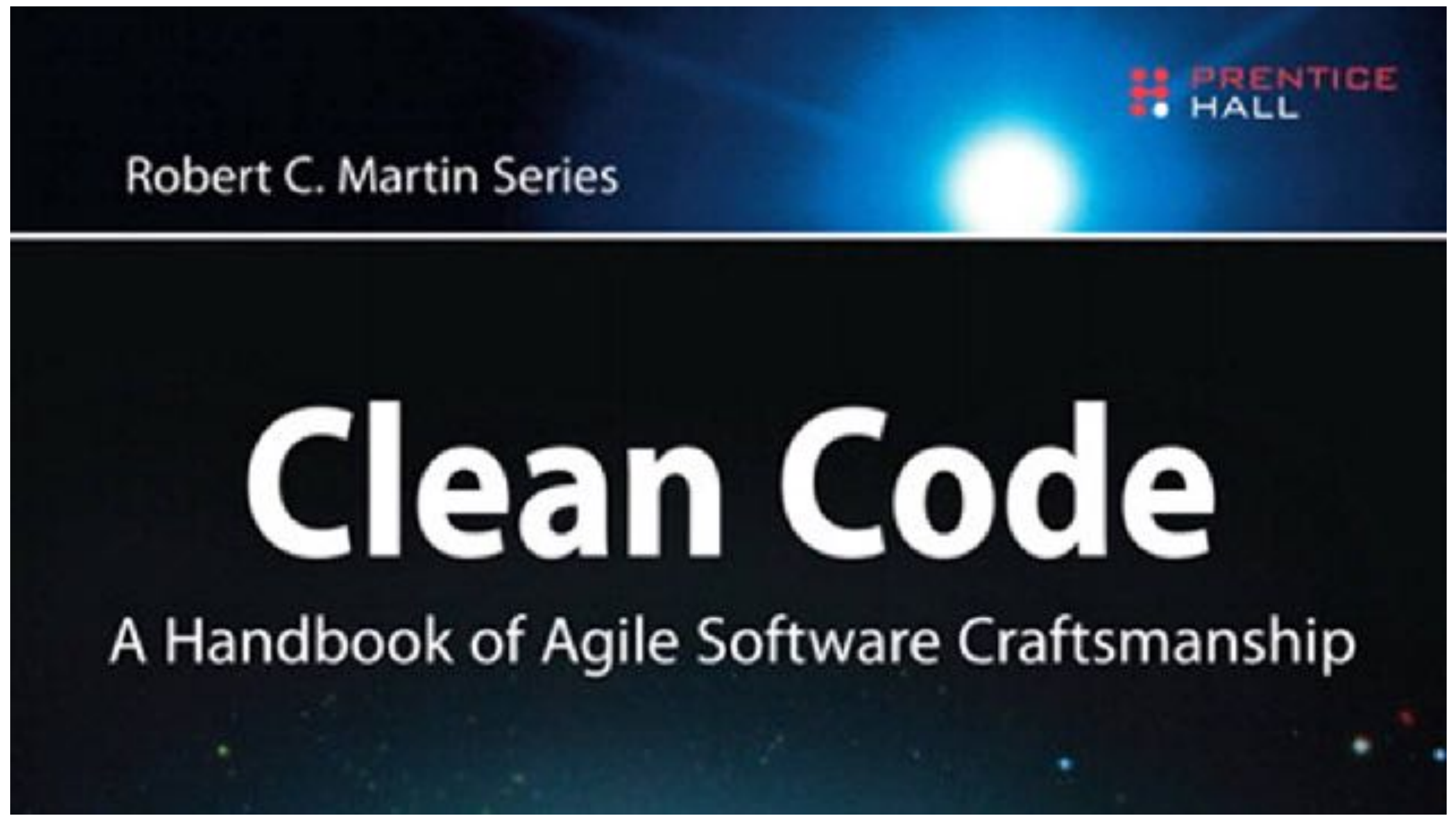
- about what the code does, not how the code changes

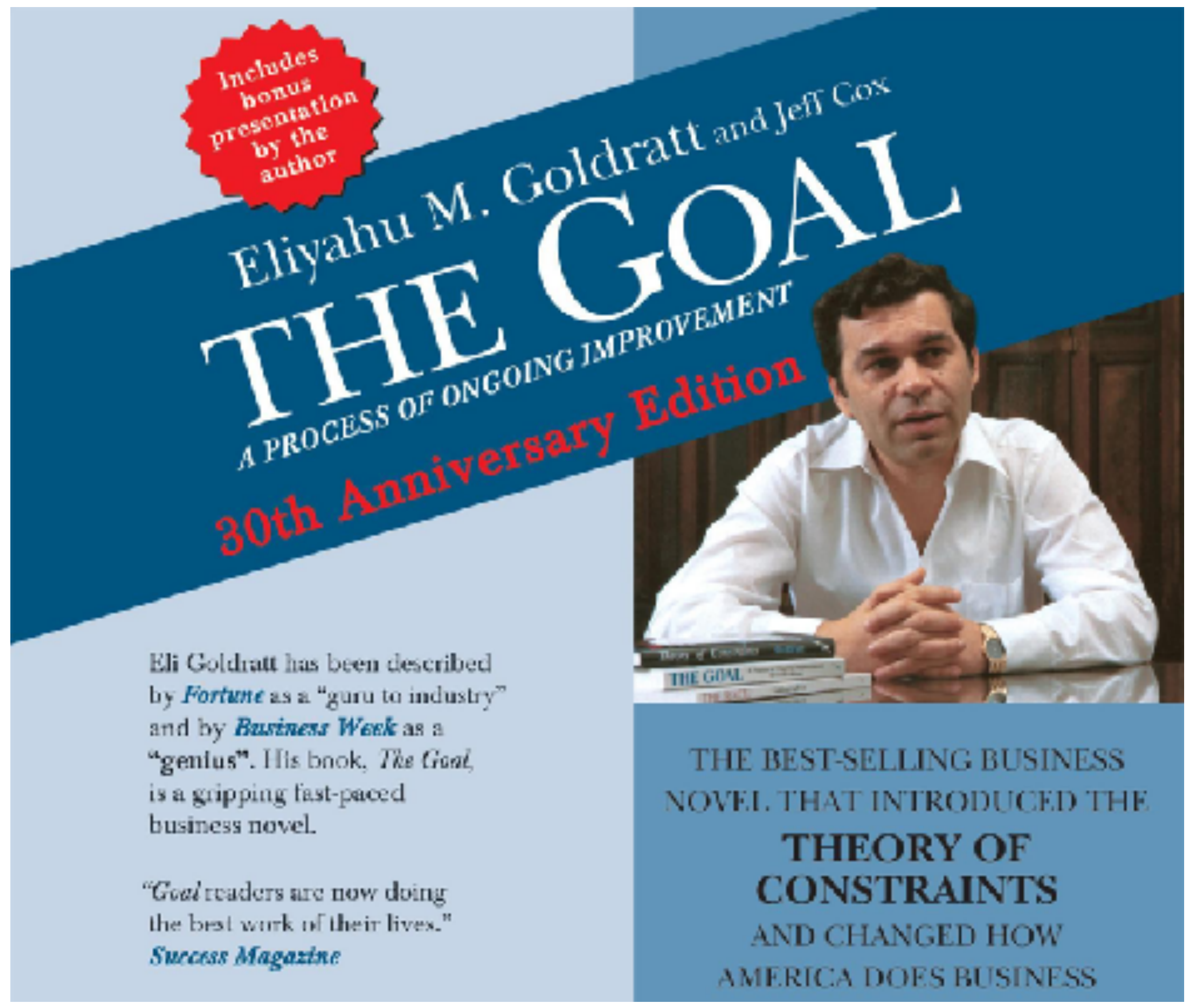
we interrupt this talk…



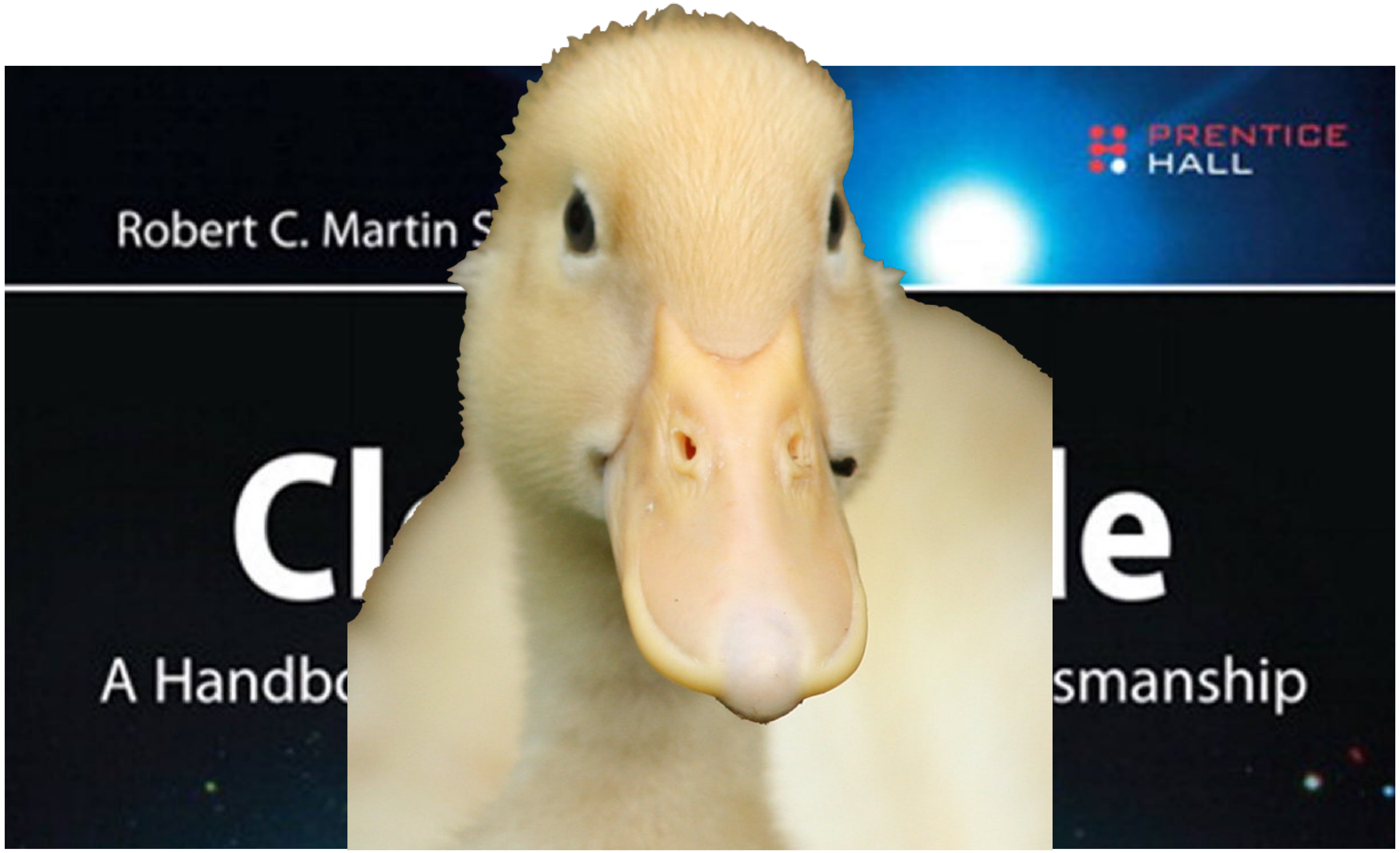
ageing like wine vs ageing like milk


A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley

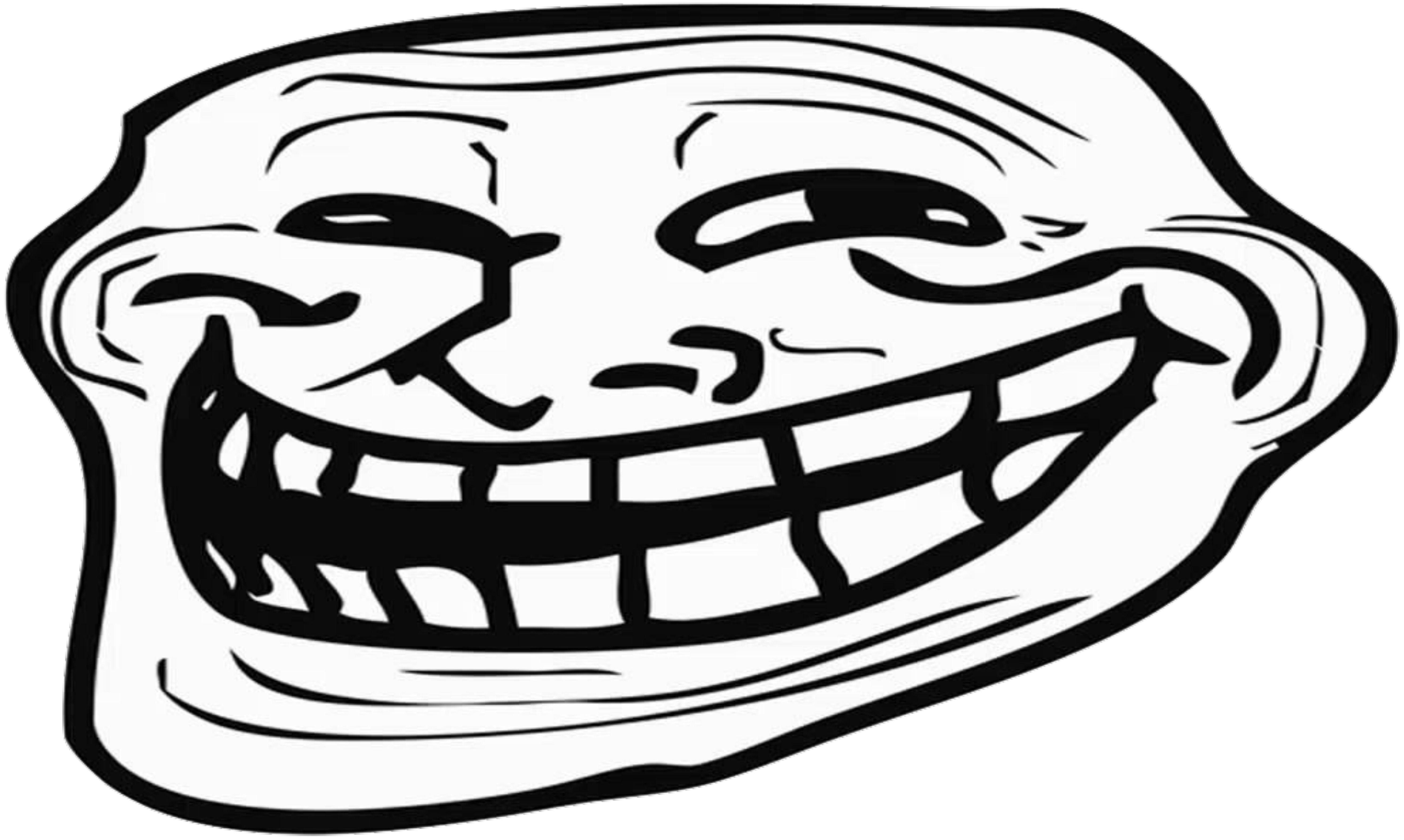



Includes bonus presentation by the author
Eliyahu M. Goldratt and Jeff Cox
THE GOAL
A Process of Ongoing Improvement
30th Anniversary Edition
Eli Goldratt has been described by Fortune as a "guru to industry" and by Business Week as a "genius". His book, The Goal, is a gripping fast-paced business novel.
"Goal readers are now doing the best work of their lives." Success Magazine
THE BEST-SELLING BUSINESS NOVEL THAT INTRODUCED THE THEORY OF CONSTRAINTS AND CHANGED HOW AMERICA DOES BUSINESS

ageing like milk



we now return to your scheduled talk…



Predictable

Behaves as expected, with no surprises

- "Passes all tests" – Kent Beck
- even when there are no tests!

Deterministic

- does the same thing every time
- well-understood operating characteristics

Observable

- in the technical sense – internal state can be inferred from outputs
- instrumentation, telemetry

Idiomatic

Uses language idioms

- standard features, constructs, libraries, frameworks, tools
- feels natural to work with, goes with the grain

Uses local idioms

- house style: coding/design standards*, or *de facto*
- aligned with project, dependencies, platforms, organisation

* caution: may not exist!

You can only write idiomatic code if you learn the idioms!

Domain-based

Uses domain language

- “Code in the language of the domain” – *97 Things* ☺
- and remember, there are multiple domains

Uses domain structure

- Code for the solution, not the framework
- payments, loans, onboarding not models, views, controllers

Uses domain boundaries

- as module boundaries, units of deployment

CUPID – for joyful coding

Composable	– plays well with others
Unix philosophy	– does one thing well
Predictable	– does what you expect
Idiomatic	– feels natural
Domain-based	– in language and structure

CUPID applied

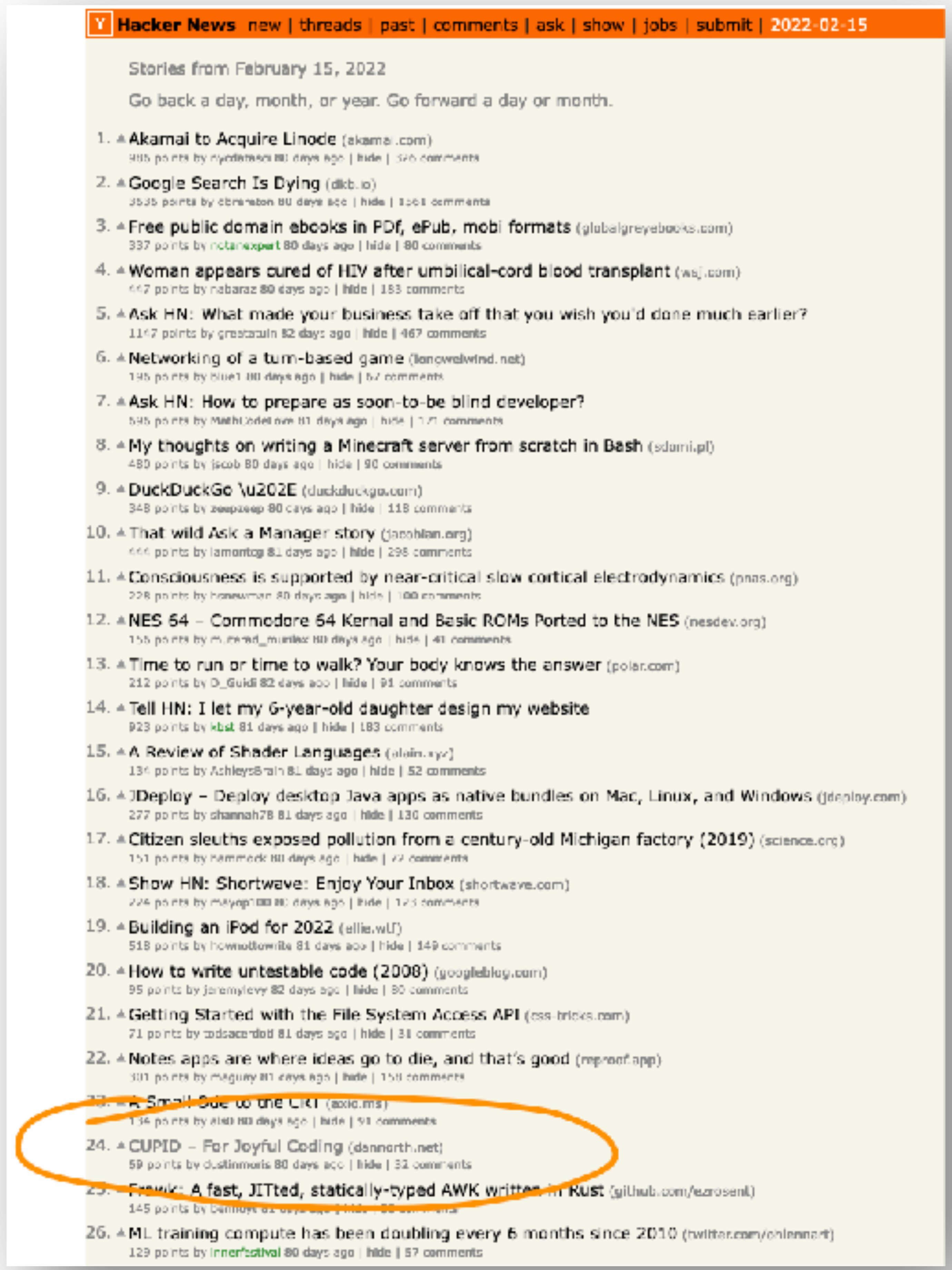
Lens for assessing a codebase

Basis for a “Code Critique”

Deciding where to start with ~~scary~~ legacy code

Syllabus for a programming course

one(ish) year later…



Hacker News new | threads | past | comments | ask | show | jobs | submit | 2022-02-15

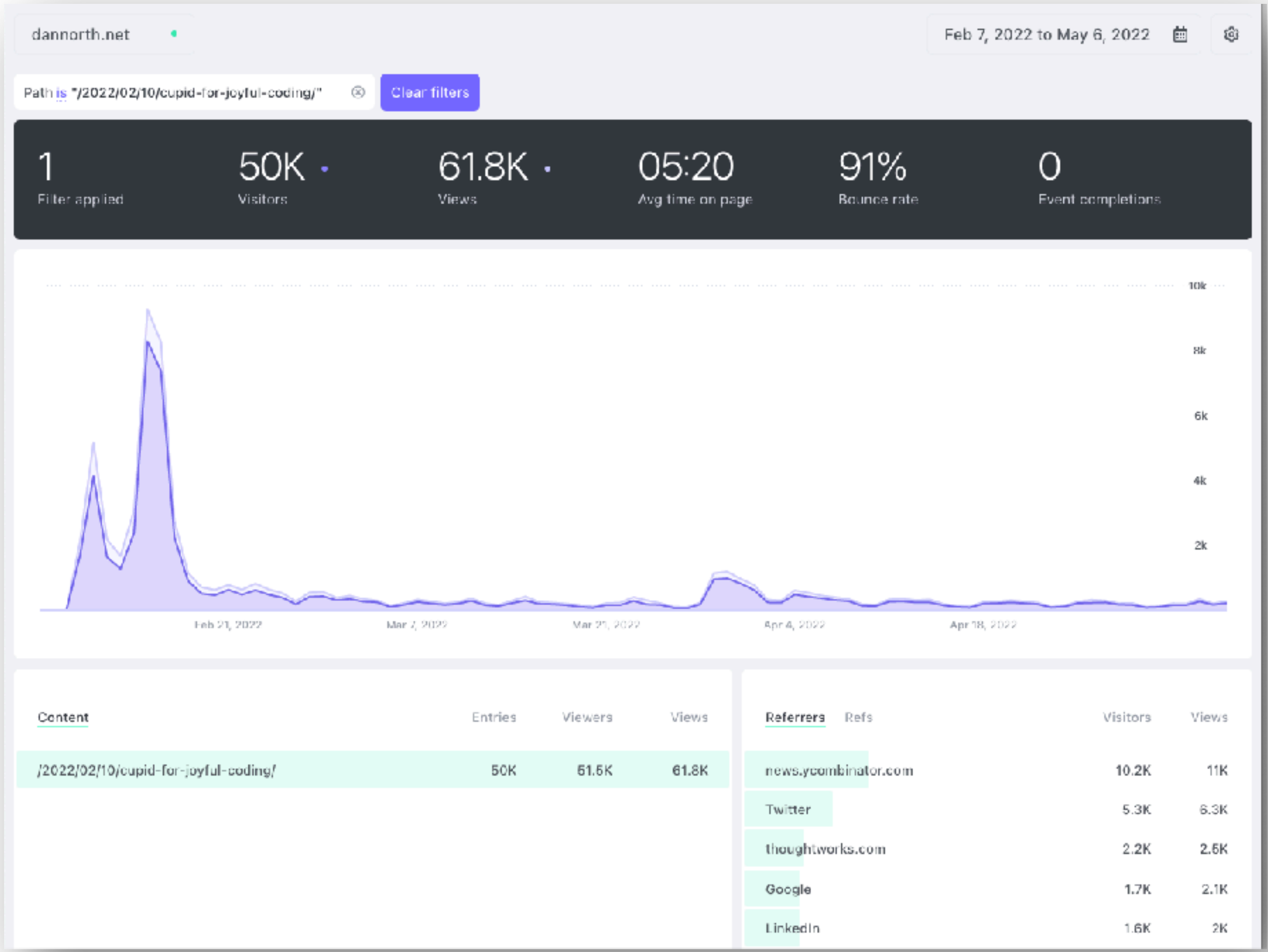
Stories from February 15, 2022

Go back a day, month, or year. Go forward a day or month.

1. ▲ Akamai to Acquire Linode (akamai.com)
   [illegible] points by [illegible] days ago | hide | [illegible] comments
2. ▲ Google Search Is Dying (dkb.io)
   3535 points by [illegible] 80 days ago | hide | [illegible] comments
3. ▲ Free public domain ebooks in PDf, ePub, mobi formats (globalgreyebooks.com)
   337 points by notarexpert 80 days ago | hide | 80 comments
4. ▲ Woman appears cured of HIV after umbilical-cord blood transplant (wsj.com)
   [illegible] points by nabaraz 80 days ago | hide | 183 comments
5. ▲ Ask HN: What made your business take off that you wish you'd done much earlier?
   1147 points by grestaulin 82 days ago | hide | 467 comments
6. ▲ Networking of a turn-based game (longwelwind.net)
   195 points by [illegible] 80 days ago | hide | 67 comments
7. ▲ Ask HN: How to prepare as soon-to-be blind developer?
   595 points by [illegible] 81 days ago | hide | [illegible] comments
8. ▲ My thoughts on writing a Minecraft server from scratch in Bash (sdomi.pl)
   480 points by jscob 80 days ago | hide | 90 comments
9. ▲ DuckDuckGo \u202E (duckduckgo.com)
   348 points by zeepzeep 80 days ago | hide | 118 comments
10. ▲ That wild Ask a Manager story (jacobian.org)
    [illegible] points by lamontcg 81 days ago | hide | 298 comments
11. ▲ Consciousness is supported by near-critical slow cortical electrodynamics (pnas.org)
    228 points by [illegible] 80 days ago | hide | 100 comments
12. ▲ NES 64 – Commodore 64 Kernal and Basic ROMs Ported to the NES (nesdev.org)
    155 points by [illegible] 80 days ago | hide | 41 comments
13. ▲ Time to run or time to walk? Your body knows the answer (polar.com)
    212 points by D_Guidi 82 days ago | hide | 91 comments
14. ▲ Tell HN: I let my 6-year-old daughter design my website
    923 points by kbst 81 days ago | hide | 183 comments
15. ▲ A Review of Shader Languages (alain.xyz)
    134 points by AshleysBrain 81 days ago | hide | 52 comments
16. ▲ JDeploy – Deploy desktop Java apps as native bundles on Mac, Linux, and Windows (jdeploy.com)
    277 points by shannah78 81 days ago | hide | 130 comments
17. ▲ Citizen sleuths exposed pollution from a century-old Michigan factory (2019) (science.org)
    151 points by [illegible] 80 days ago | hide | [illegible] comments
18. ▲ Show HN: Shortwave: Enjoy Your Inbox (shortwave.com)
    224 points by [illegible] 80 days ago | hide | [illegible] comments
19. ▲ Building an iPod for 2022 (ellie.wtf)
    518 points by [illegible] 81 days ago | hide | 149 comments
20. ▲ How to write untestable code (2008) (googleblog.com)
    95 points by jeremylevy 82 days ago | hide | 80 comments
21. ▲ Getting Started with the File System Access API (css-tricks.com)
    71 points by todsacerdoti 81 days ago | hide | 31 comments
22. ▲ Notes apps are where ideas go to die, and that's good (reproof.app)
    301 points by [illegible] 81 days ago | hide | 158 comments
23. ▲ A Small Ode to the CRT (axio.ms)
    134 points by [illegible] 80 days ago | hide | 91 comments
24. ▲ CUPID – For Joyful Coding (dannorth.net)
    59 points by dustinmoris 80 days ago | hide | 32 comments
25. ▲ Frawk: A fast, JITted, statically-typed AWK written in Rust (github.com/ezrosent)
    145 points by [illegible]
26. ▲ ML training compute has been doubling every 6 months since 2010 (twitter.com/ohlennart)
    129 points by innerfestival 80 days ago | hide | 57 comments

@tastapod

spiky traffic is spiky!



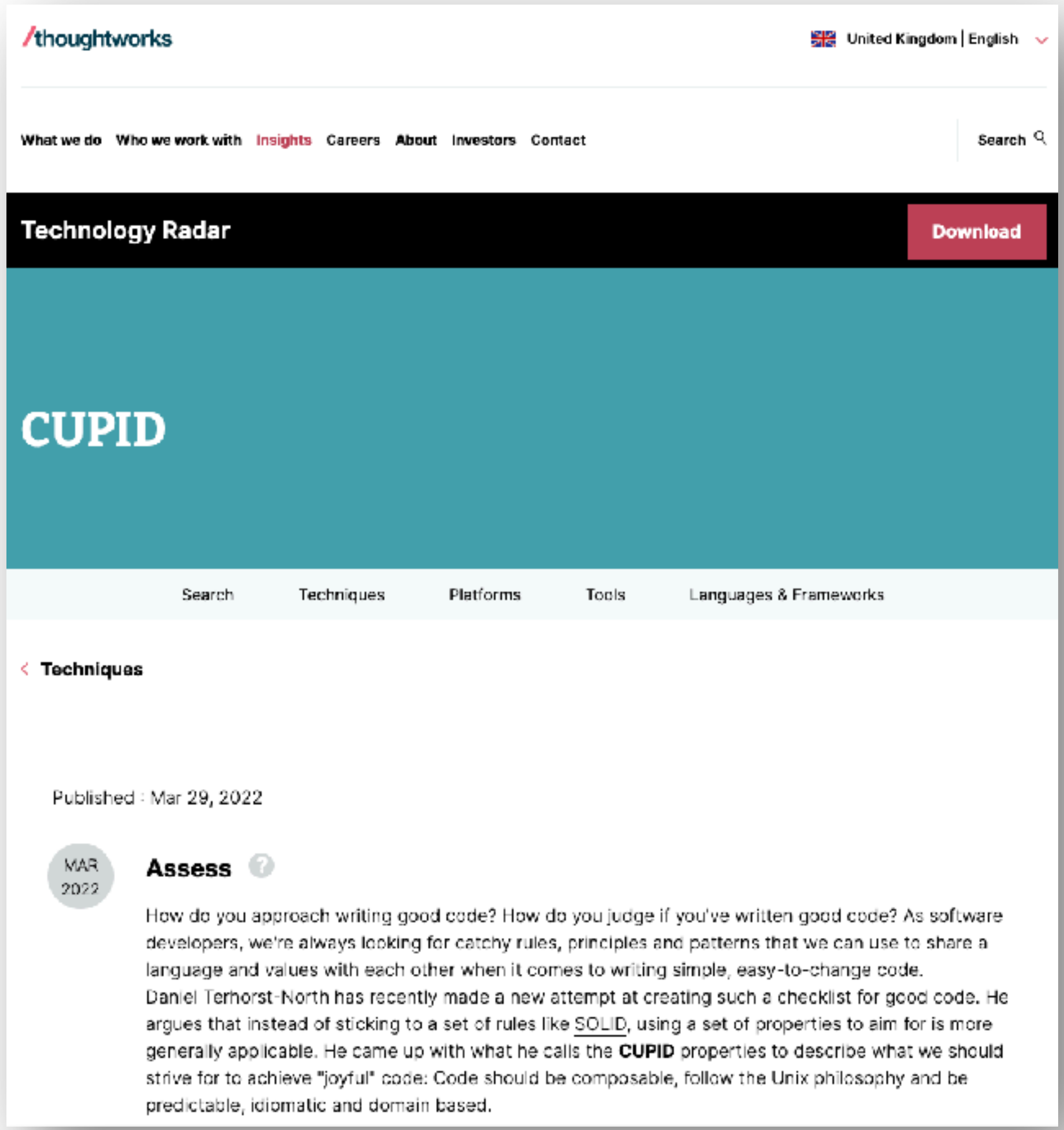
1	50K	61.8K	05:20	91%	0
Filter applied	Visitors	Views	Avg time on page	Bounce rate	Event completions

Content	Entries	Viewers	Views
/2022/02/10/cupid-for-joyful-coding/	50K	61.5K	61.8K

Referrers Refs	Visitors	Views
news.ycombinator.com	10.2K	11K
Twitter	5.3K	6.3K
thoughtworks.com	2.2K	2.5K
Google	1.7K	2.1K
LinkedIn	1.6K	2K

@tastapod

literally on the radar



Published : Mar 29, 2022

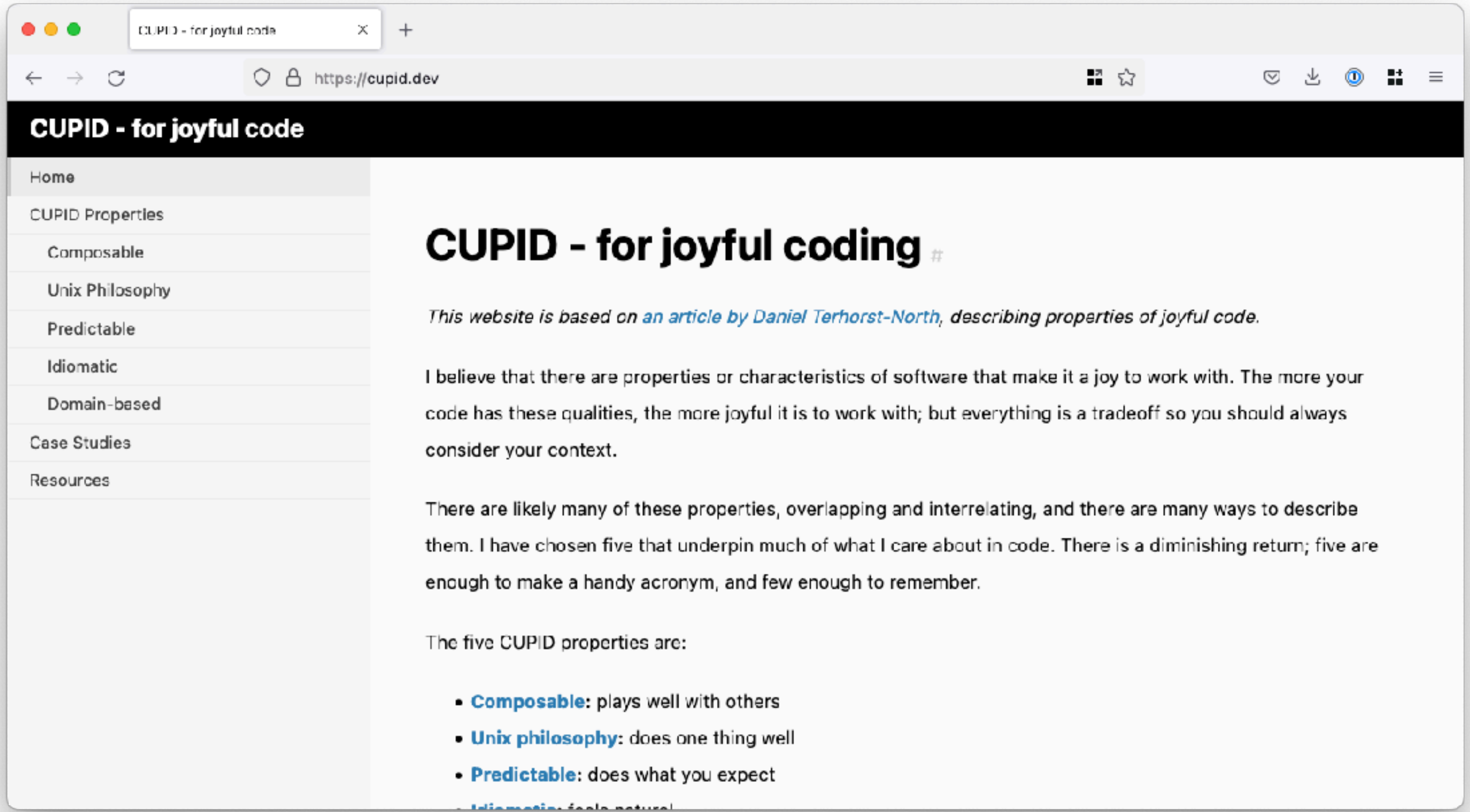
MAR 2022

Assess

How do you approach writing good code? How do you judge if you've written good code? As software developers, we're always looking for catchy rules, principles and patterns that we can use to share a language and values with each other when it comes to writing simple, easy-to-change code.
Daniel Terhorst-North has recently made a new attempt at creating such a checklist for good code. He argues that instead of sticking to a set of rules like SOLID, using a set of properties to aim for is more generally applicable. He came up with what he calls the **CUPID** properties to describe what we should strive for to achieve "joyful" code: Code should be composable, follow the Unix philosophy and be predictable, idiomatic and domain based.

@tastapod

...and we're live! – cupid.dev



CUPID - for joyful code

https://cupid.dev

CUPID - for joyful code

- Home
- CUPID Properties
  - Composable
  - Unix Philosophy
  - Predictable
  - Idiomatic
  - Domain-based
- Case Studies
- Resources

CUPID - for joyful coding

This website is based on an article by Daniel Terhorst-North, describing properties of joyful code.

I believe that there are properties or characteristics of software that make it a joy to work with. The more your code has these qualities, the more joyful it is to work with; but everything is a tradeoff so you should always consider your context.

There are likely many of these properties, overlapping and interrelating, and there are many ways to describe them. I have chosen five that underpin much of what I care about in code. There is a diminishing return; five are enough to make a handy acronym, and few enough to remember.

The five CUPID properties are:

- **Composable**: plays well with others
- **Unix philosophy**: does one thing well
- **Predictable**: does what you expect

Join us at https://groups.io/g/cupid-joyful-code

@tastapod

share your stories!

https://groups.io/g/cupid-joyful-code

https://dannorth.net

daniel@dannorth.net