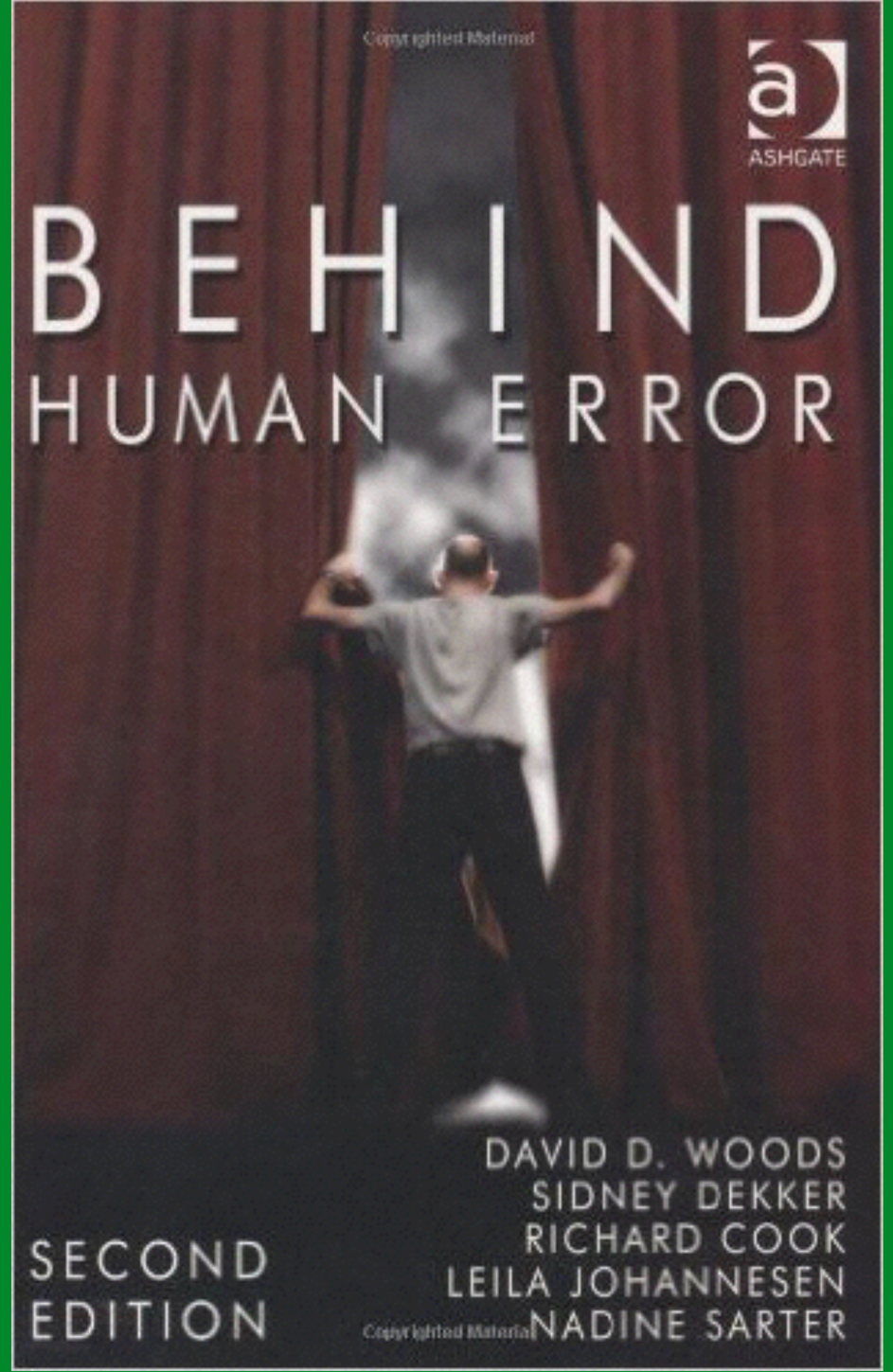
Lessons From Web Operations

or

“Software, software, everywhere…and not a human to think.”

John Allspaw
Etsy



This is NOT optional.

Future

- Wherefore art thou *tacit knowledge?*
- Finding **adaptive cycles**
- Exploring how software engineers *reason* about their systems and code

- 2010-present, CTO, Etsy.com
- 2015, Master's Program, HF/Systems Safety, Lund University
- Author of chapter on web ops in "HF/E in Practice" http://bit.ly/1OkgLPP

Etsy

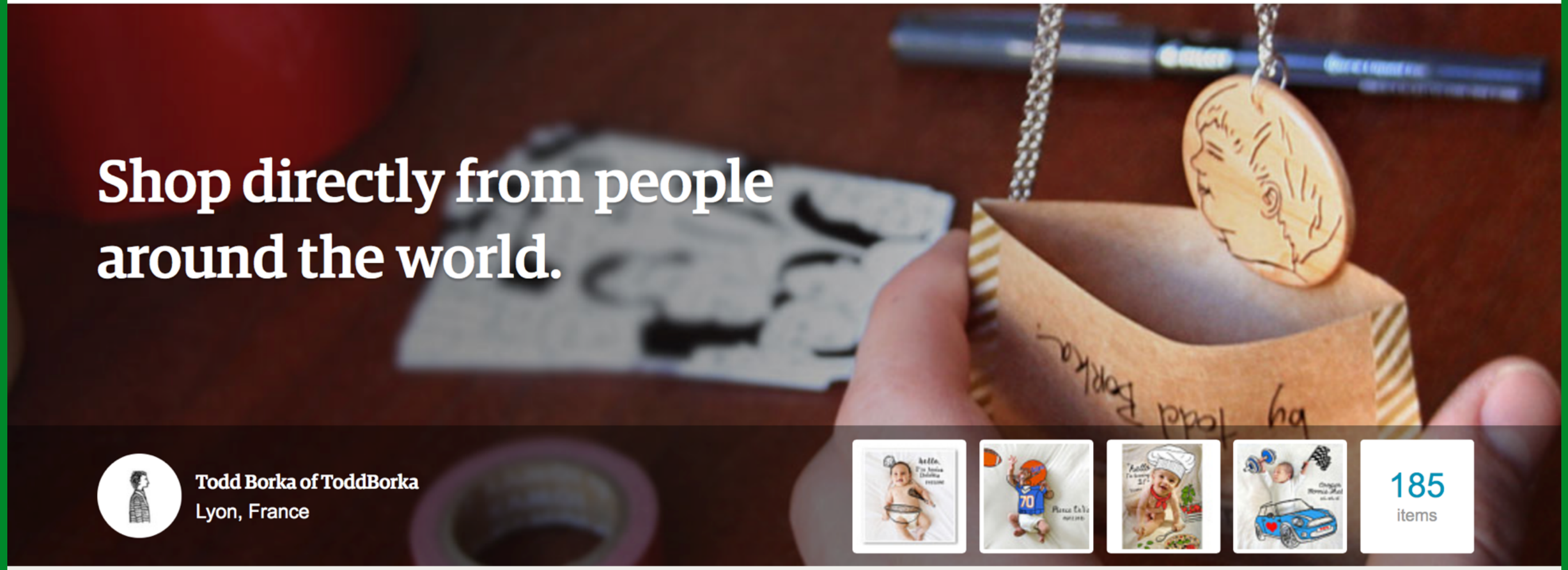
Search for items or shops

Search

Sell on Etsy Register Sign in

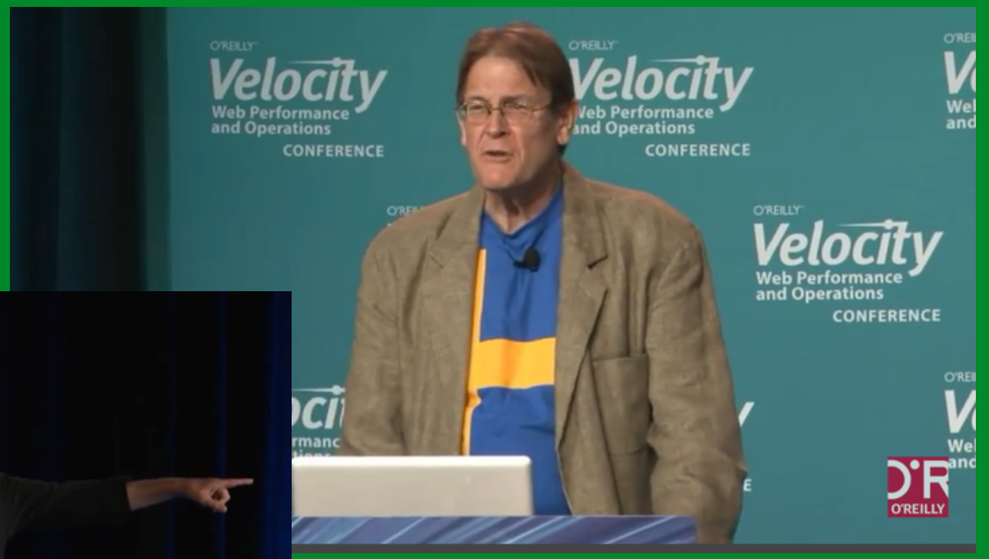
Cart

Clothing & Accessories Jewelry Craft Supplies & Tools Weddings Entertainment Home & Living Kids & Baby Vintage

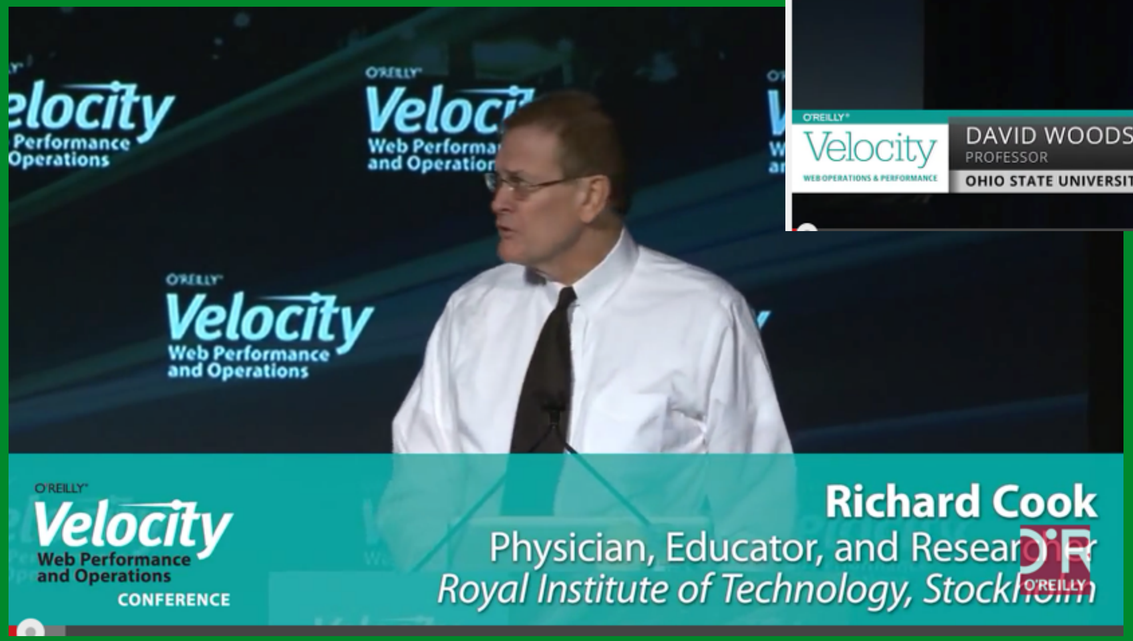


Discover items you can't find anywhere else


Velocity
Web Performance
and Operations
CONFERENCE
O'REILLY


Velocity
Web Performance
and Operations
CONFERENCE
O'REILLY


Velocity
WEB OPERATIONS & PERFORMANCE
DAVID WOODS
PROFESSOR
OHIO STATE UNIVERSITY
O'REILLY


Velocity
Web Performance
and Operations
CONFERENCE
Richard Cook
Physician, Educator, and Researcher
Royal Institute of Technology, Stockholm
O'REILLY


O'REILLY

INDEPENDENT

News Voices Culture Lifestyle Tech Sport

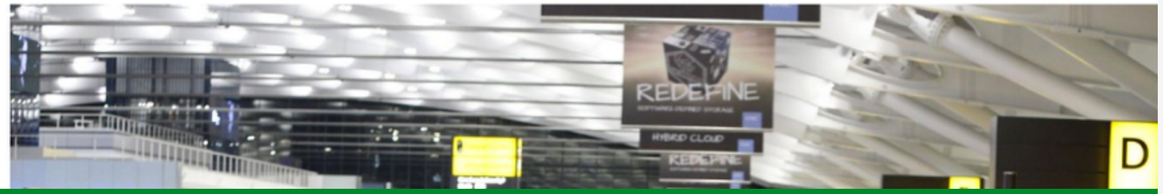
Home News

London-wide air traffic control failure was caused by an unprecedented computer fault

Air traffic controllers workload was increased by the error, which was deemed unsafe

Jon Stone | @joncstone | Saturday 13 December 2014 15:29 GMT | 2 comments

0 shares



theguardian

home › tech UK world sport football opinion culture business lifestyle all

Software

Software upgrade grounds hundreds of flights over US east coast

FAA says 'technical issues' with an air traffic control computer undergoing a software update caused 492 flight delays and 476 cancellations over weekend

Embracing Millenial
@samphippen

Follow

The tech community: we drink because computers are awful.

RETWEETS
2

FAVORITES
4

2:13 PM - 28 Mar 2015

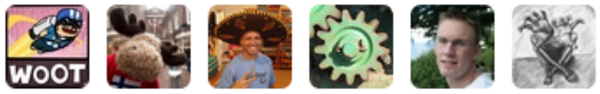
Jérôme Petazzoni
@jpetazzo

Follow

Computers are terrible. Nothing works. But it's still the closest we'll ever get to being actual wizards unwiredcouch.com/2014/05/21/com … — by @mrtazz

RETWEETS
4

FAVORITES
2



9:28 AM - 2 Jul 2014

Ethan Marcotte
@beep

Follow

never use computers, computers are terrible, always avoid computers

RETWEETS 44 FAVORITES 77

8:58 AM - 1 Dec 2014

Nope Francis
@sorenmacbeth

Follow

C.R.E.A.M - computers ruin everything around me

RETWEETS	FAVORITES
72	89

7:52 PM - 5 Oct 2015

fiona
@fioroco

Following

the more you know about computers, the less afraid of robots you are! up until you know lot about computers

RETWEETS 5 FAVORITES 6

5:59 AM - 6 Oct 2015

Software Operations

“efficiently implementing automated abstractions”

(Guo, 2010).

Software Operations

"*All* companies are software companies."

SMTP

- 220 Billion emails sent per day
- 120 per day sent by average business user


who let you do this

Personal details of world leaders accidentally revealed by G20 organisers

Exclusive: Obama, Putin, Merkel, Cameron, Modi and others kept in the dark after passport numbers and other details were disclosed in Australia's accidental privacy breach

- Follow our full coverage of this exclusive story
- Read the immigration department's letter outlining the circumstances of the G20 privacy breach

Tony Abbott and Vladimir Putin cuddle koalas before the start of the first G20 meeting in November 2014.

False alarm: Iranians' sudden access to Facebook and Twitter a tech glitch

Published time: September 17, 2013 15:54

Get short URL

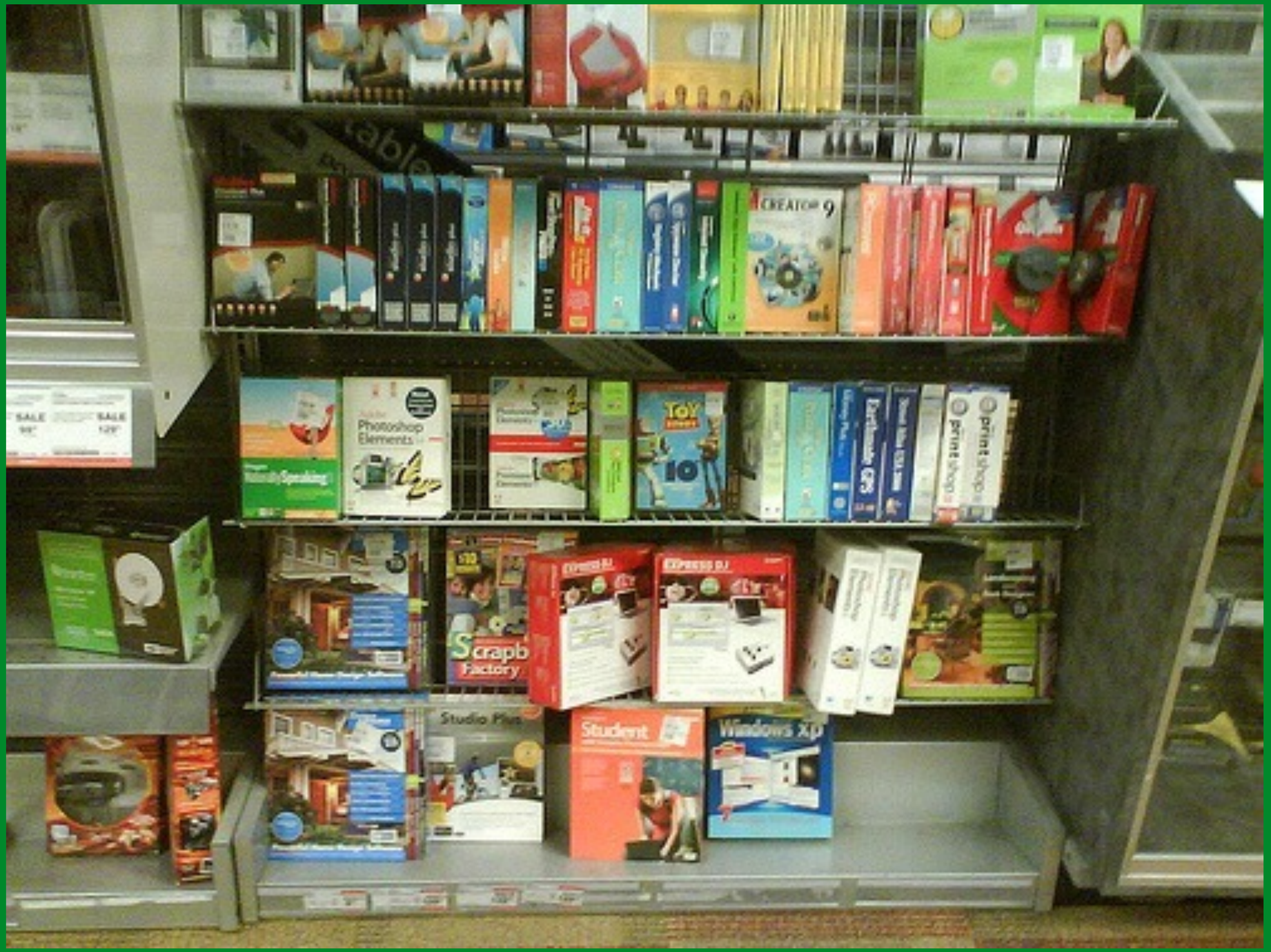
Iran's President Hassan Rohani (Reuters)

592 129 1

Facebook and Twitter users in Iran unexpectedly got unhindered access to US-based social networks sparking hopes that the blocks had been removed – only to find several hours later that it was no more than a computer glitch.

Tags

Information Technology, Internet, Iran, Law, Security, Social networks


Photoshop Elements
TOY STORY 10
Scrapb Factory
Studio Plus
Student
Windows xp

Dynamic Fault Management

Asinine Algorithmic Assumptions

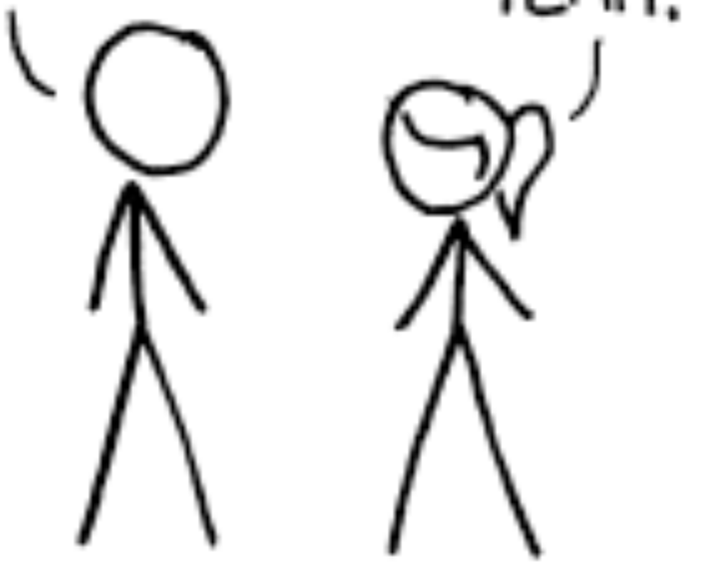
Testing?


INSPECTED BY
204
for the greater glory of our capitalist overlords.

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

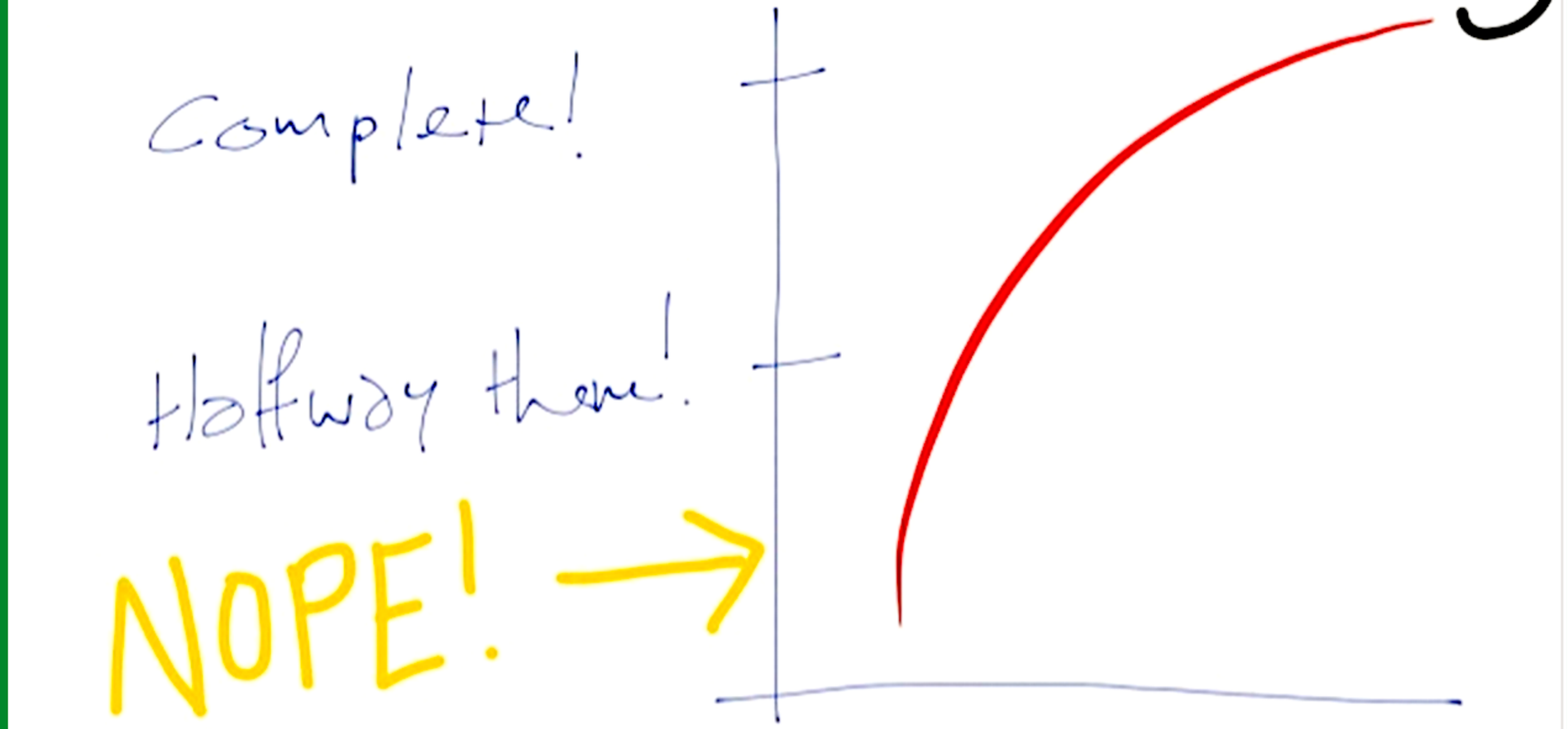
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.

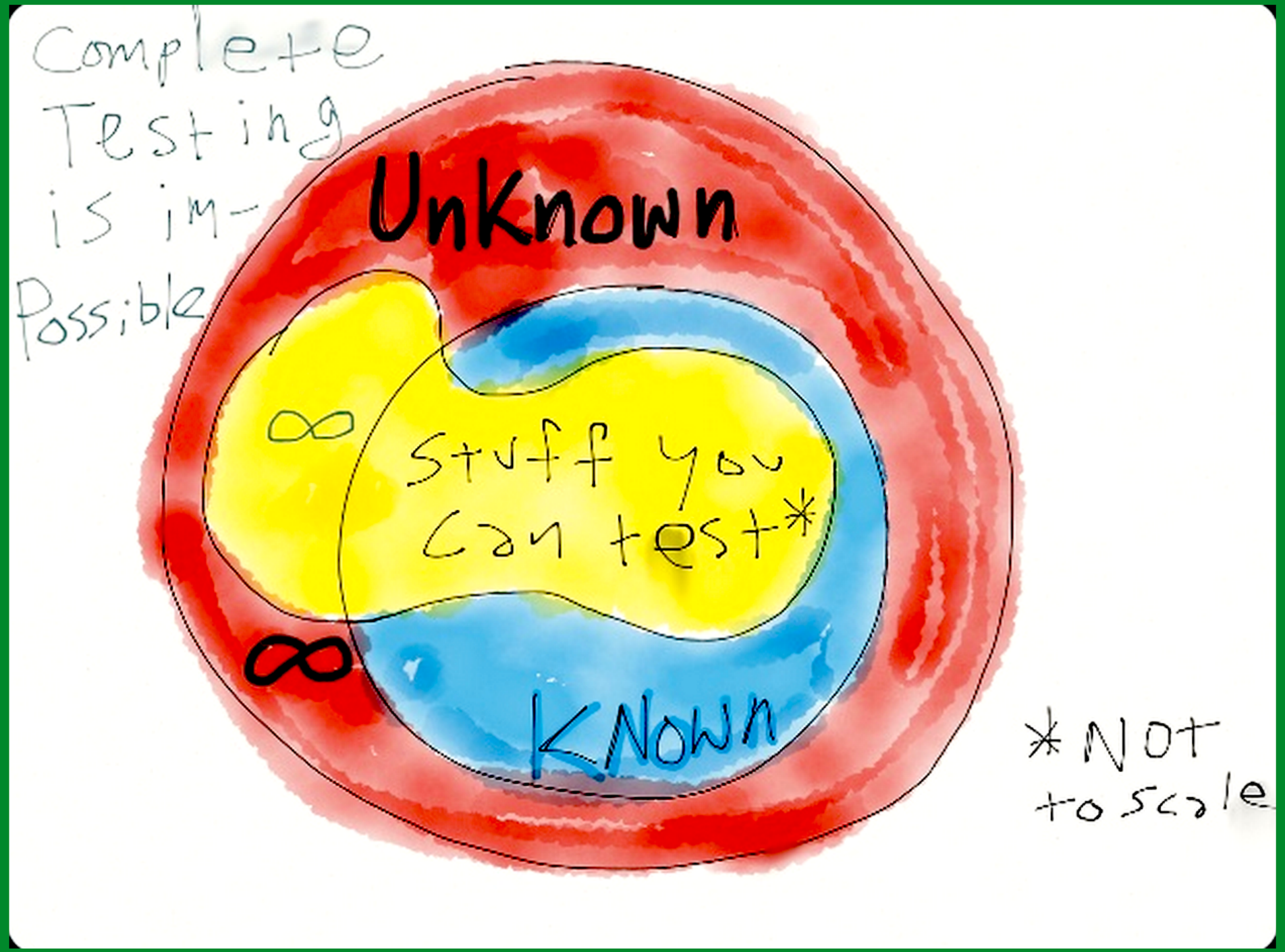


SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

We do not asymptotically approach complete testing




Complete
Testing
is im-
Possible
Unknown
∞
Stuff you
Can test*
∞
KNOWN
*NOT
to scale

Fallacies of distributed computing

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

Falsehoods Programmers Believe About Time

1. There are always 24 hours in a day.
2. Months have either 30 or 31 days.
3. Years have 365 days.
4. February is always 28 days long.
5. If a process runs for n seconds and then terminates, approximately n seconds will have elapsed on the system clock at the time of termination.
6. Time always goes forwards. The system clock will always be set to the correct local time.
7. ...

79.

http://infiniteundo.com/post/25326999628/falsehoods-programmers-believe-about-time

User as Operator as Author as Designer

Vast majority of software and architecture in ***critical business path*** are adapted/adjusted/modified by the people who design it and use it.


IT WORKED FINE IN TEST
OPS PROBLEM NOW

Data (“big” or “little”)

- Our ability to record+collect data about our systems outstrip our ability to make sense of it all.
- Trivial to collect, very difficult (if not impossible) to comprehend
- All-too-familiar topics:
    - data overload/underload
    - directed attention
    - alert design

“No plan survives first contact with the enemy.”

“No ~~plan~~ survives first contact with ~~the enemy~~.”

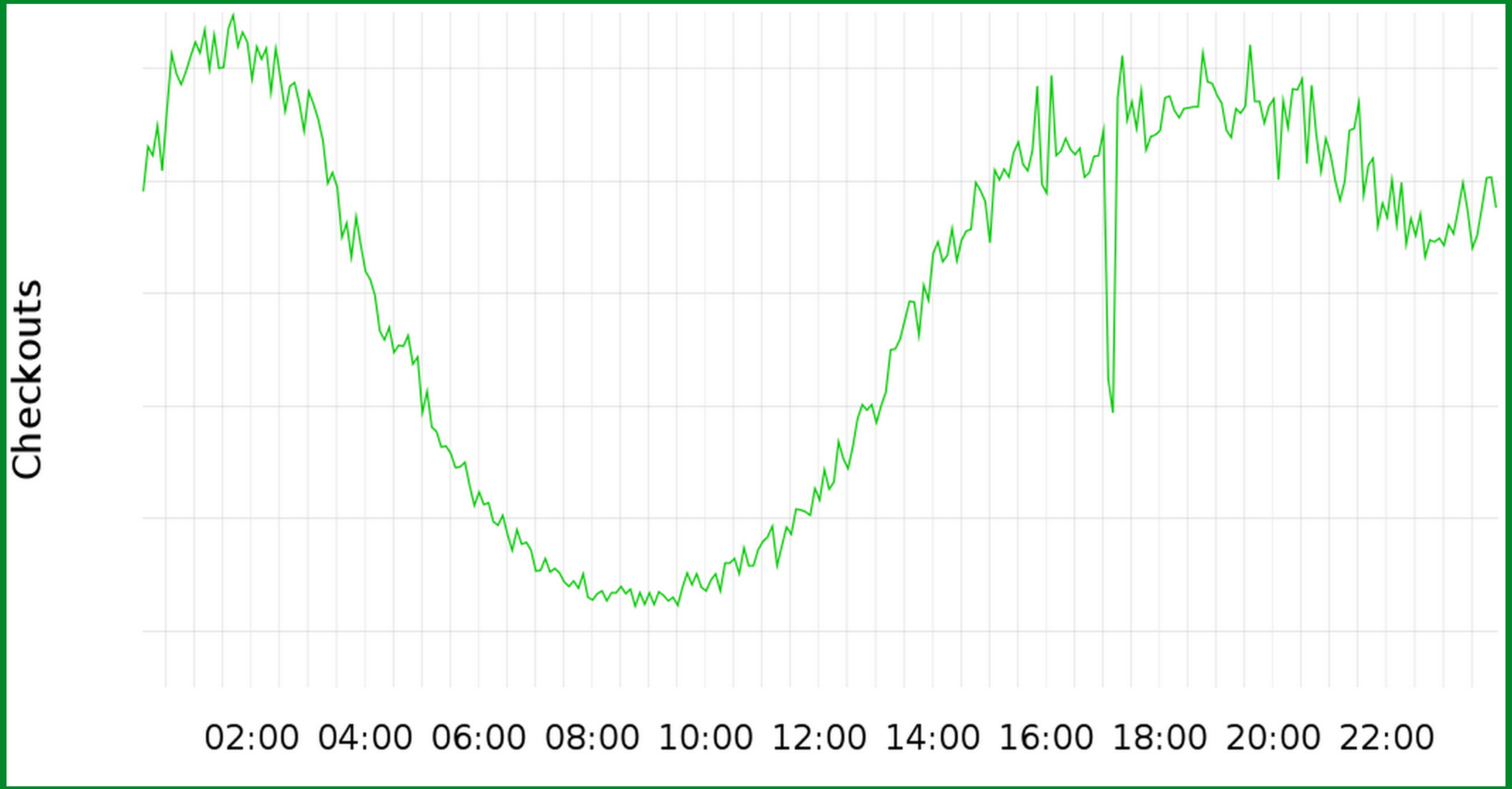
^ ^

“perfect” software production traffic.

A Story

nonce

A Story


Checkouts
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00

A Story

Etsy

Code as Craft

Speaker Series Events About Archive

Blameless PostMortems and a Just Culture

Posted by **John Allspaw** on May 22, 2012

Last week, Owen Thomas wrote a flattering article over at Business Insider on how we handle errors and mistakes at Etsy. I thought I might give some detail on how that actually happens, and why.

Anyone who's worked with technology at any scale is familiar with failure. Failure cares not about the architecture designs you slave over, the code you write and review, or the alerts and metrics you meticulously pore through.

So: failure happens. This is a foregone conclusion when working with complex

j.mp/BlamelessPostmortems

Hey everybody -

Don’t do what I did. I tried to do X, but because I didn’t know about Y, it was no good.

It almost exploded everything.

So, don’t do: *(details about X)*

Love,
Joe

p.s. I changed X so that Y shouldn’t matter. Can someone sanity check my work?

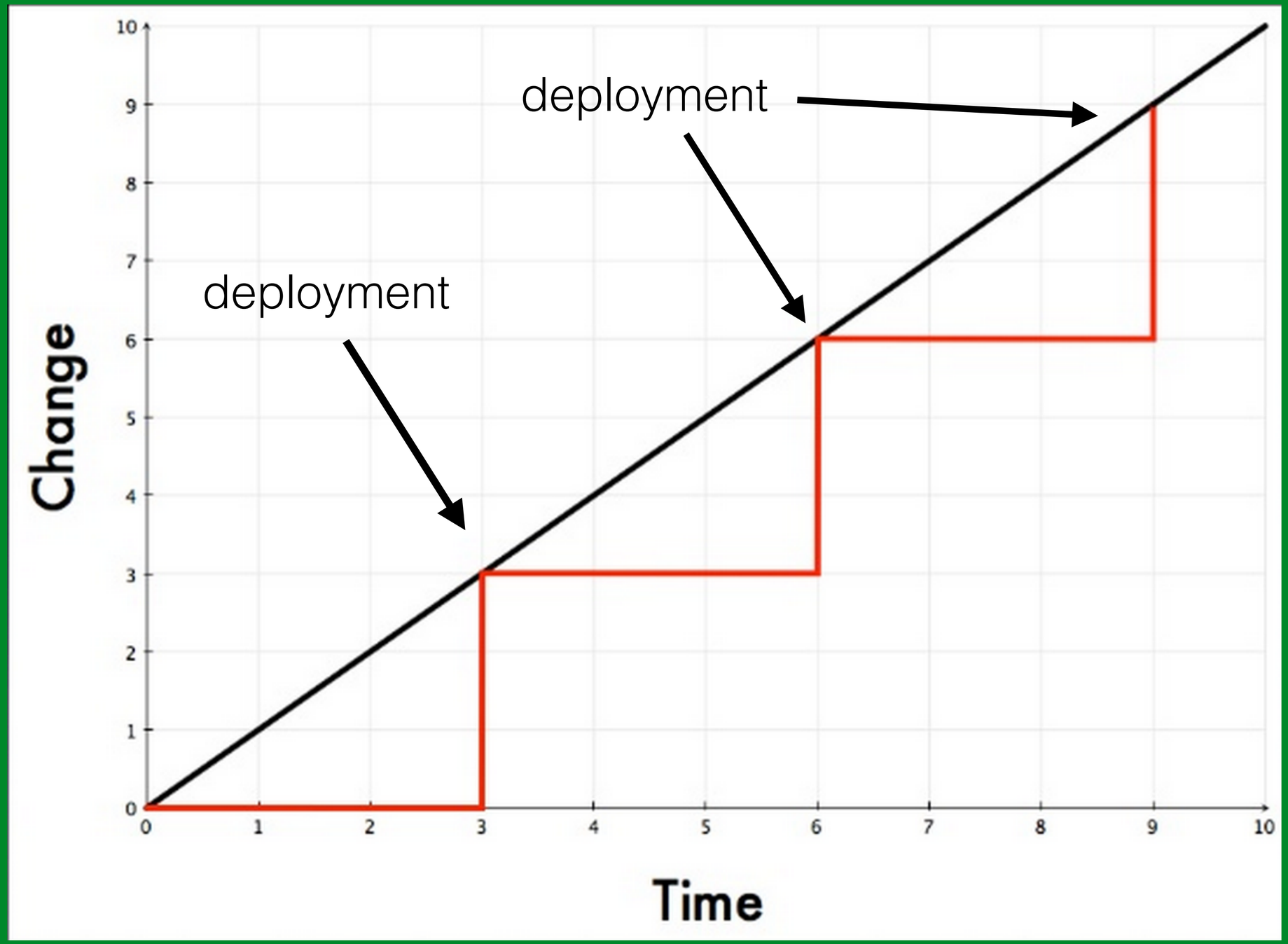
Adaptation

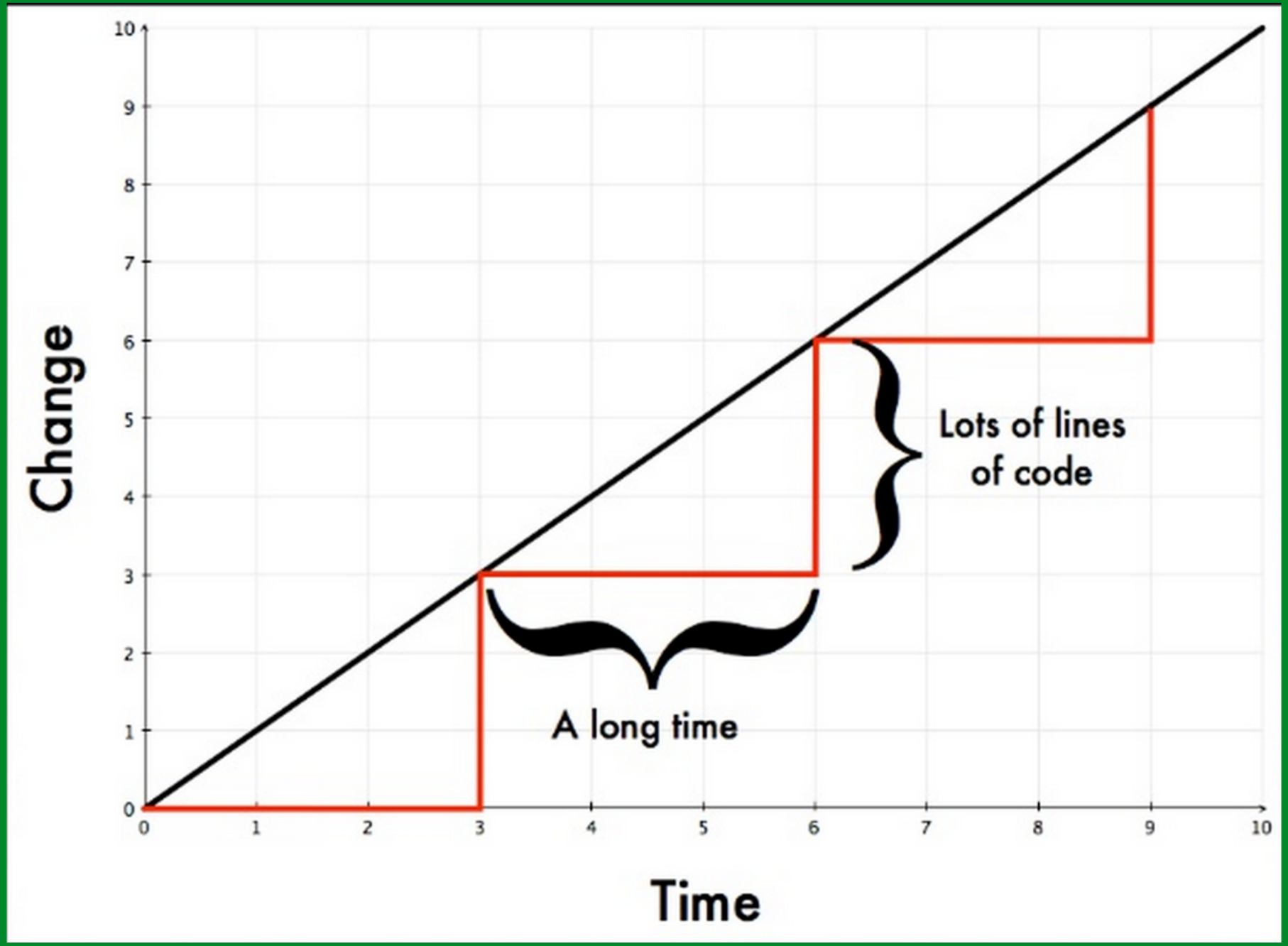
Horizontal Exchange

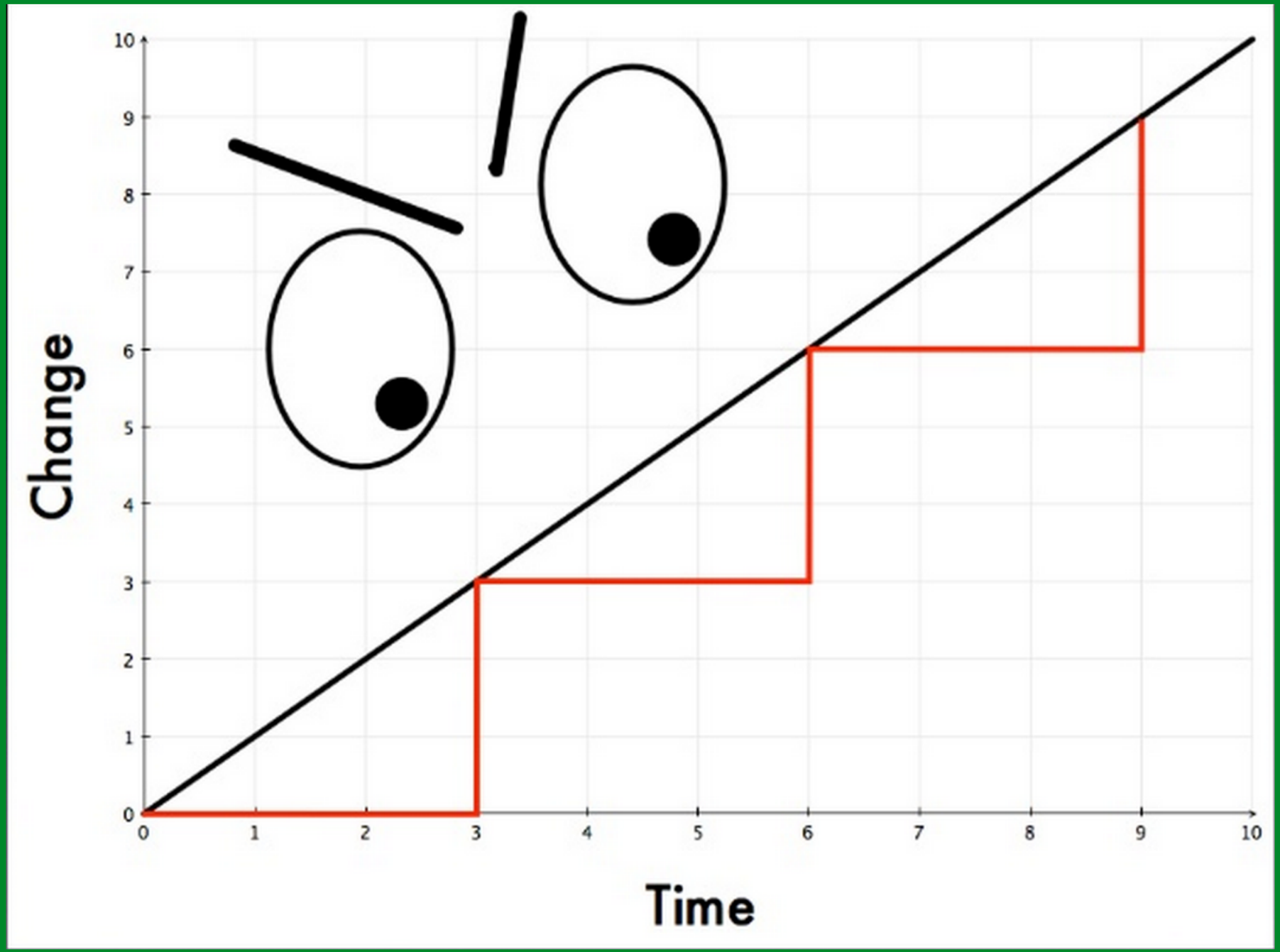
- Bootcamp and annual rotations
- Support rotations
- First Push Program
- Cross-checking, PostMortems, and PSAs

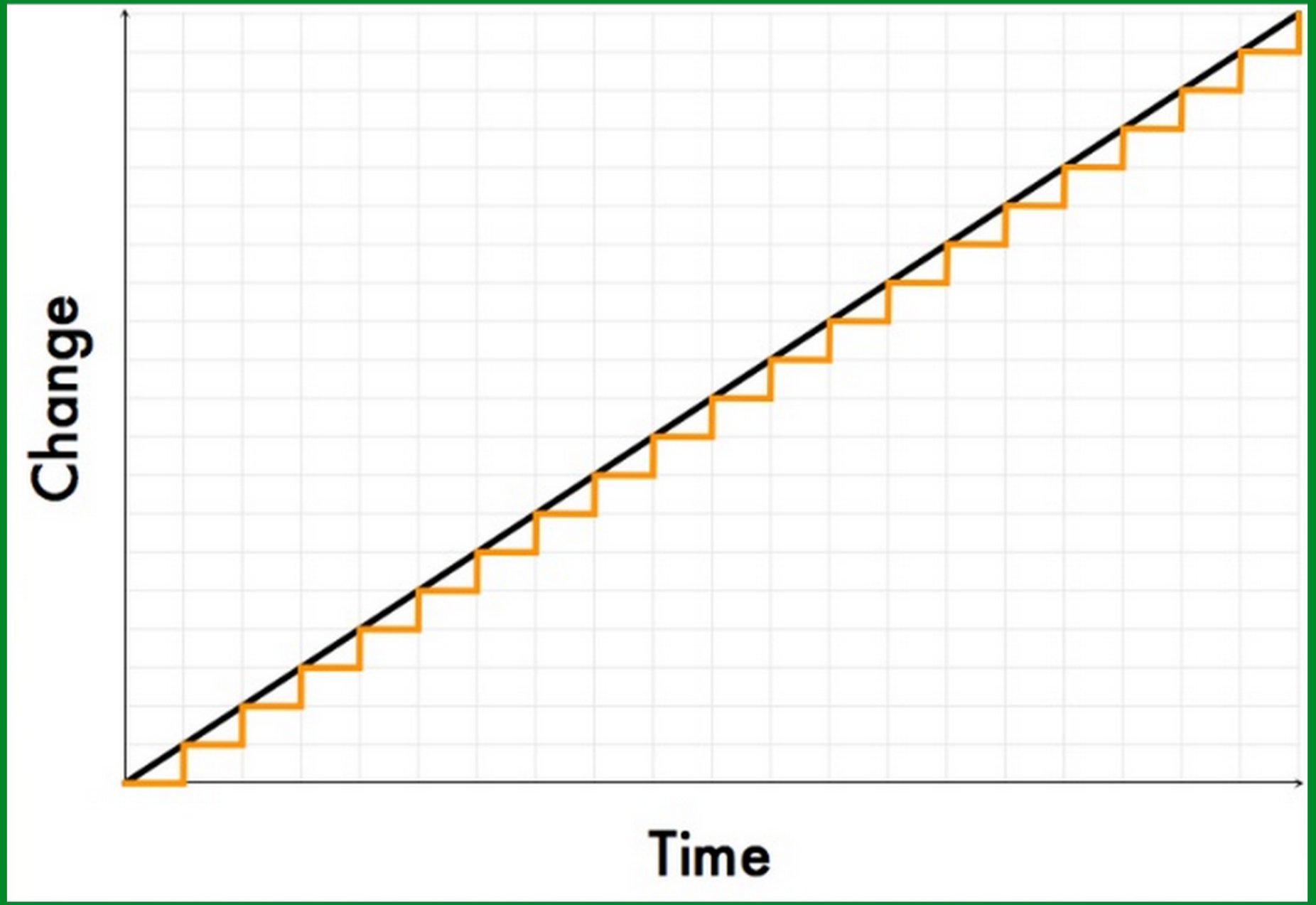
Confidence

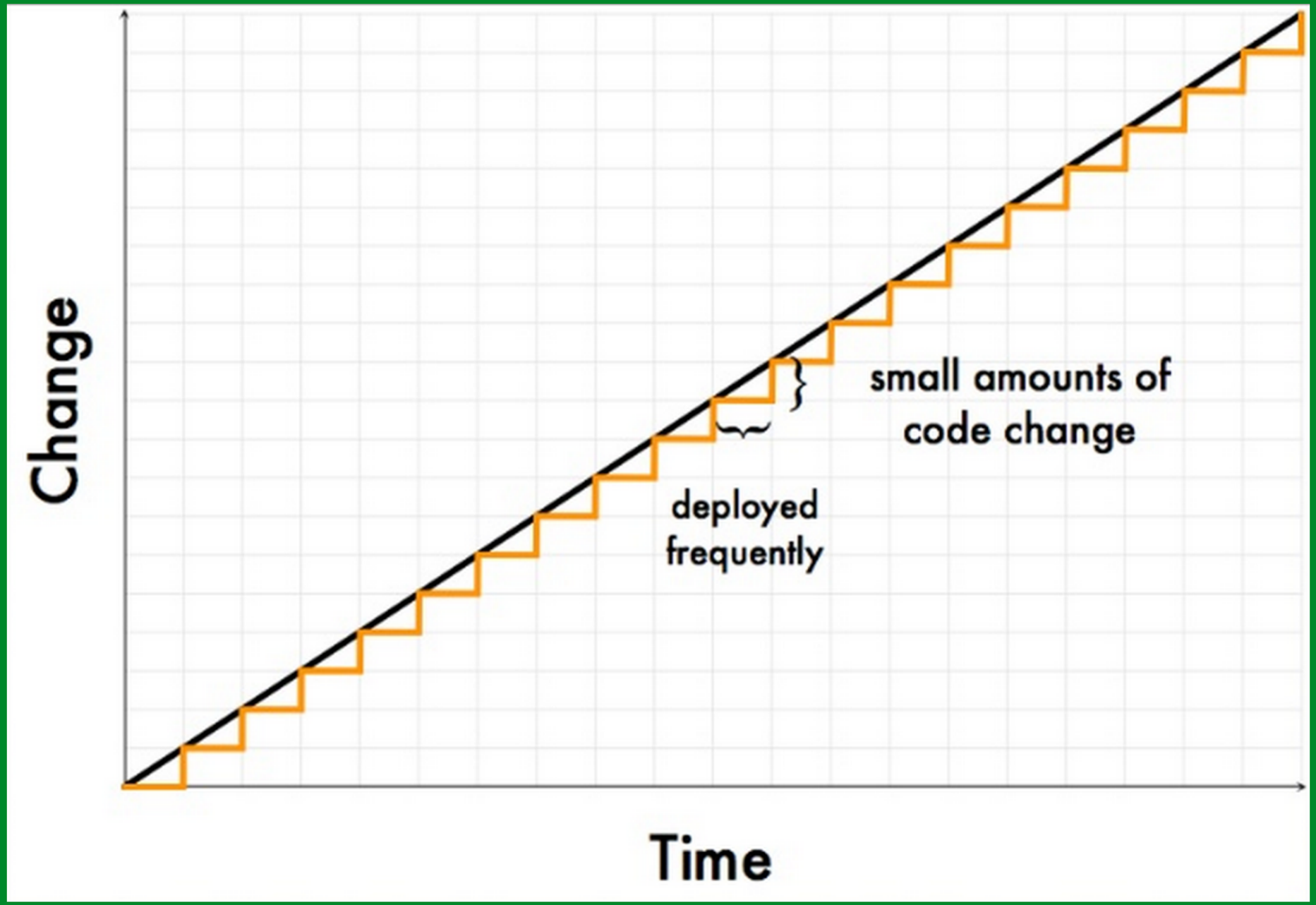
- Automated testing
- Dark deployments
- % Rampups
- Staff-only
- Peer code review
- Exploratory testing

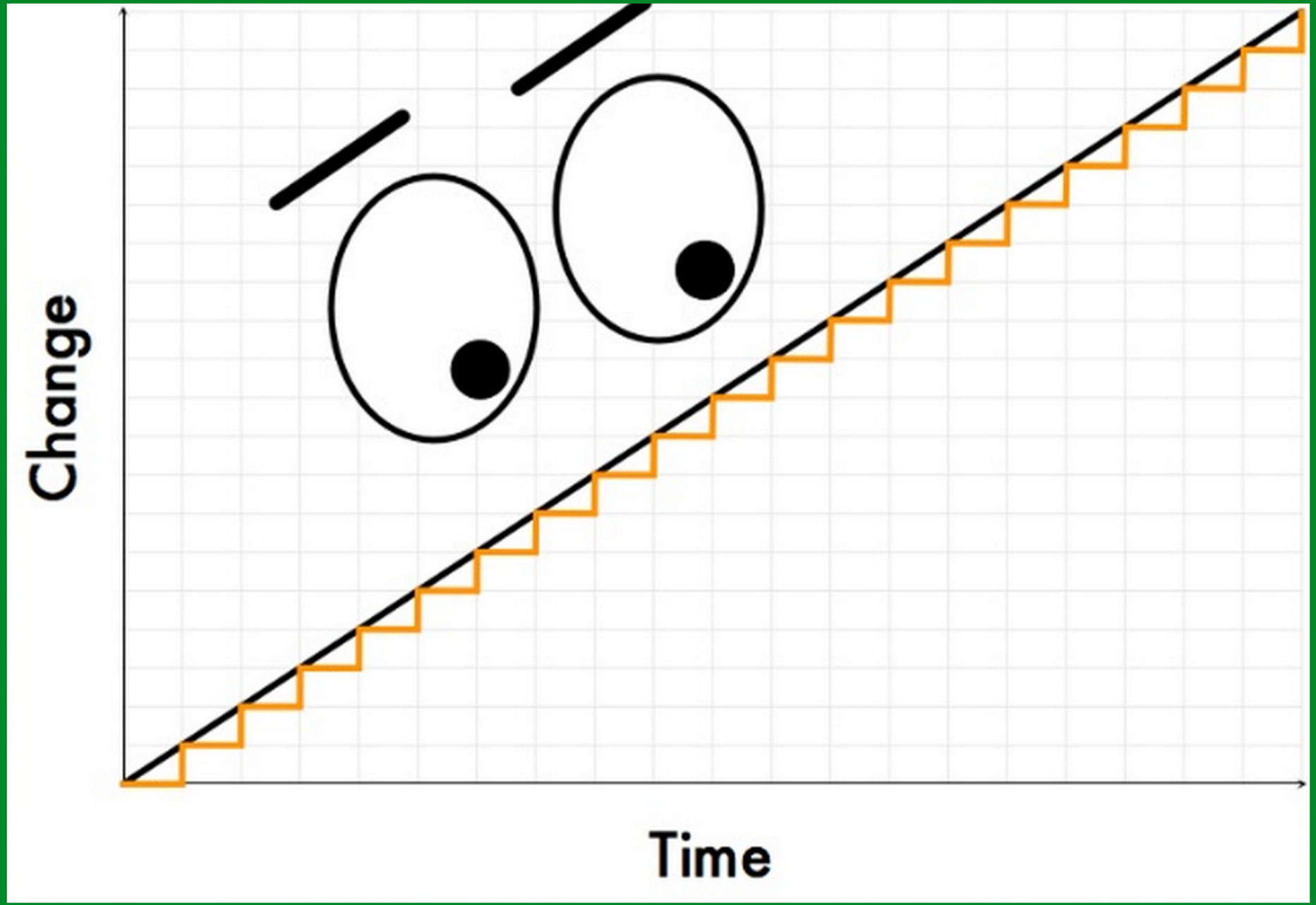

deployment
deployment
Change
Time
0
1
2
3
4
5
6
7
8
9
10


Change
Time
Lots of lines
of code
A long time
0
1
2
3
4
5
6
7
8
9
10

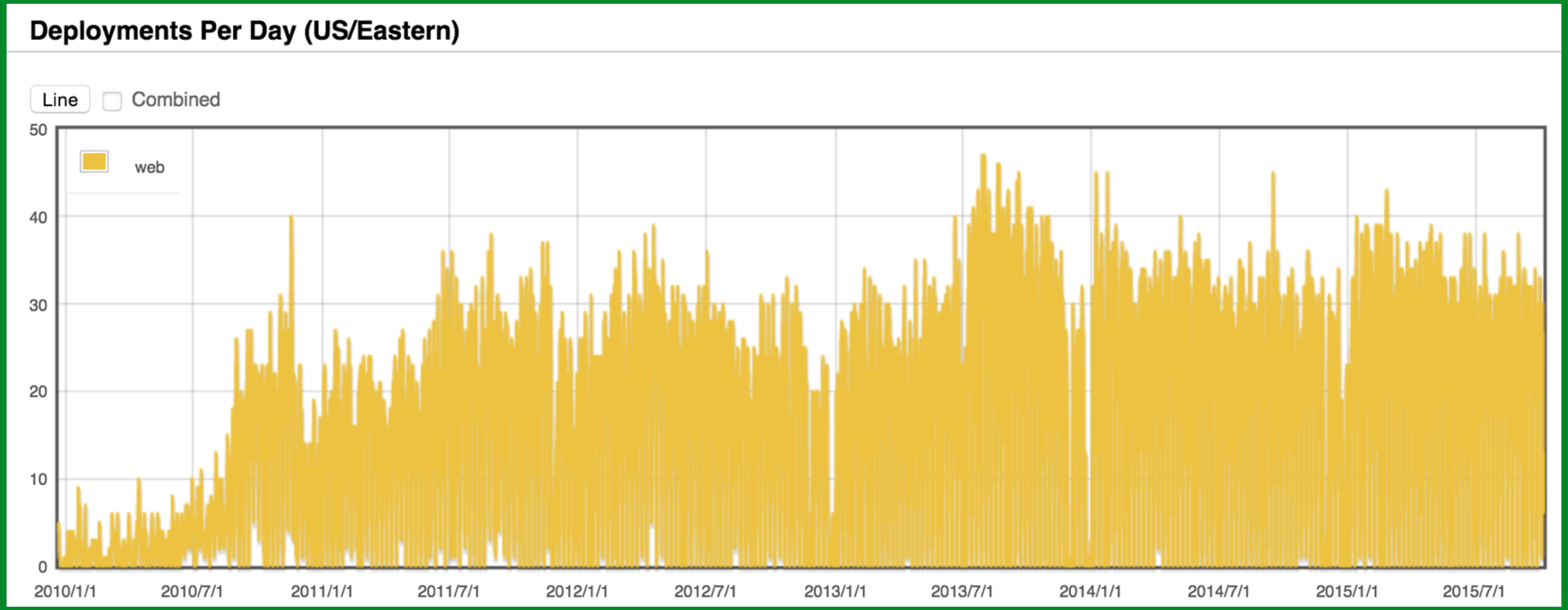

Change
Time
0
1
2
3
4
5
6
7
8
9
10


Change
Time


Change
small amounts of
code change
deployed
frequently
Time


Change
Time

Continuous Deployment



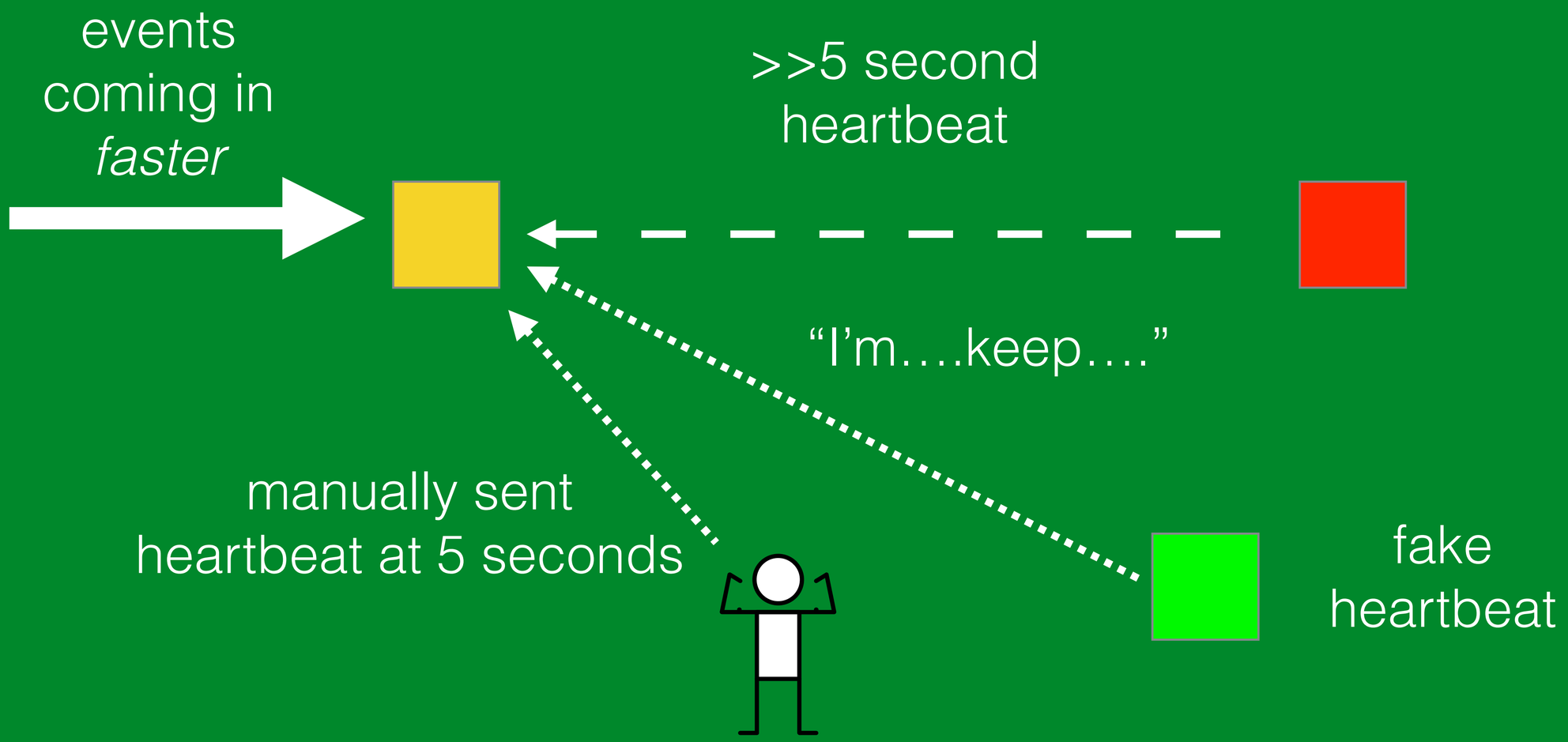
Make Change-Making “Cheap”

- Feature/config flags
- Small changes done frequently > large changes done rarely
- Admission that all changes are not created equal
- Admission that there’s no such thing as deterministic software

CASE: Trading Exchange

“I’m healthy, keep it coming…”

CASE: Trading Exchange



CASE: Etsy

- Backfilled data in databases to repair a (relatively) minor formatting issue
- Script went wrong (but fast!) and corrupted data
- Went to Hadoop (BigData) stack for recovery

acmqueue

Fault Injection in Production

Making the case for resilience testing

John Allspaw, Etsy

When we build Web infrastructures at Etsy, we aim to make them resilient. This means designing them carefully so that they can sustain their (increasingly critical) operations in the face of failure. Thankfully, there have been a couple of decades and reams of paper spent on researching how fault tolerance and graceful degradation can be brought to computer systems. That helps the cause.

To make sure that the resilience built into Etsy systems is sound and that the systems behave as expected, we have to see the failures being tolerated *in production*.

Why production? Why not simulate this in a QA or staging environment? First, the existence of any differences in those environments brings uncertainty to the exercise, and second, the risk of not recovering has no consequences during testing, which can bring hidden assumptions into the fault-tolerance design and into recovery. The goal is to reduce uncertainty, not increase it.

Forcing failures to happen, or even designing systems to fail on their own, generally isn't easily sold to management. Engineers are not conditioned to embrace their ability to respond to emergencies; they aim to avoid them altogether. Taking a detailed look at how to respond better to failure is essentially accepting that failure will happen, which you might think is counter to what you want in engineering, or in business.

Future For Software Operations

- Wherefore art thou ***tacit knowledge***?
- Finding more **adaptive cycles+patterns**
- Exploring how software engineers ***reason*** about their systems and code

Adaptation is *normal.*

Adaptation cannot be proceduralized.

Adaptation is really hard to see.

The End