



GOTO
Guide

LET US HELP YOU

Ask questions
through **the app**



also remember to rate session

 Download on the
App Store

 GET IT ON
Google Play

THANK YOU!

#GOTOams

SHUT UP AND EAT YOUR VEG

forgotten lessons from the 1990's (or
before)

garth



 garth.gilmour@instil.co

 @GarthGilmour

 Garth Gilmour

INSTIL



 eamonn.boyle@gearset.com

 @BoyleEamonn

 Eamonn Boyle



Eamonn Boyle

Development Manager

in computer science
we ignore our own history



the 20 year gap

you learn this as history



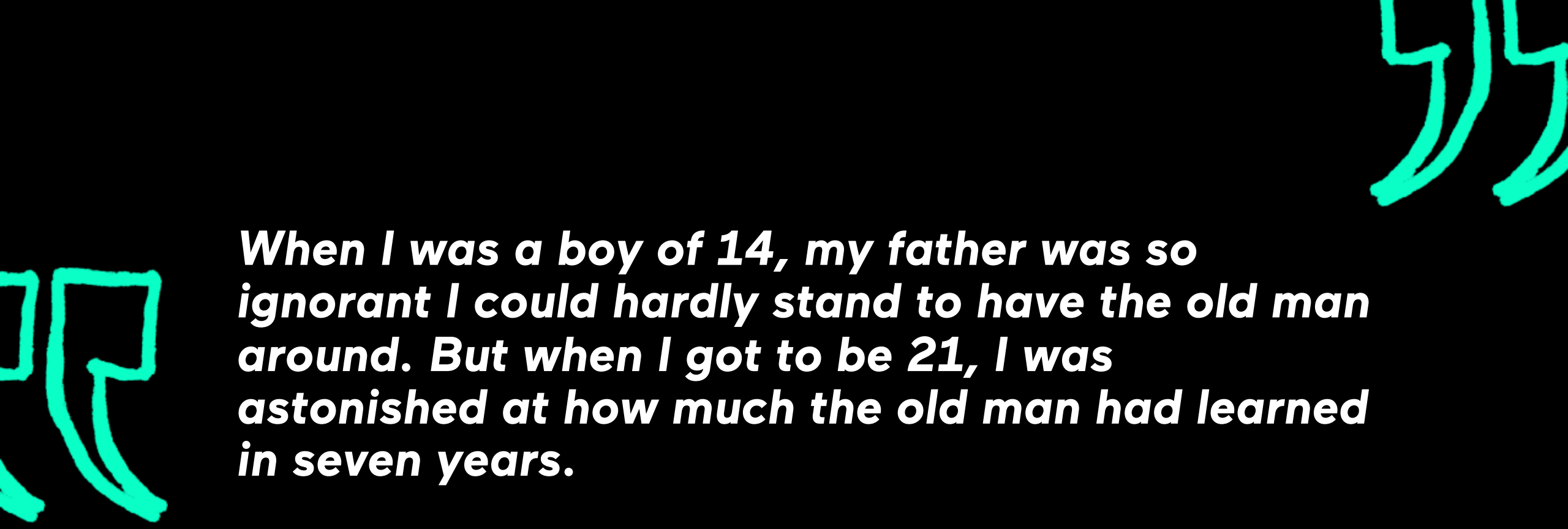
you experience this yourself

the 20 year gap

in the 1990s we were all idiots
and then they invented agile

IT'S A BIT MORE
COMPLICATED THAN
THAT





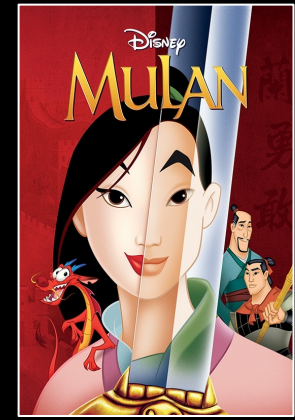
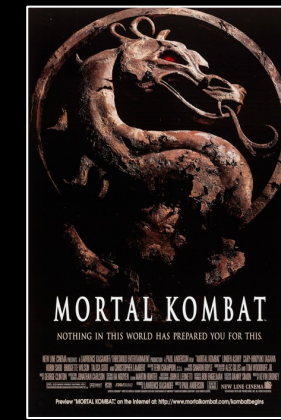
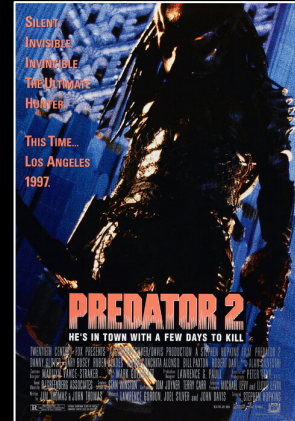
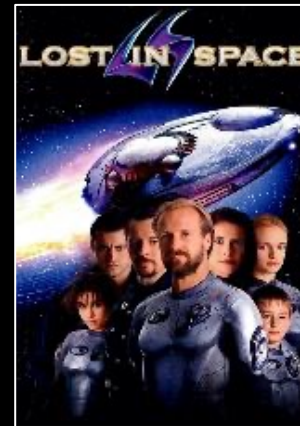
When I was a boy of 14, my father was so ignorant I could hardly stand to have the old man around. But when I got to be 21, I was astonished at how much the old man had learned in seven years.

Mark Twain

the 20 year gap

the 1990s were a crazy time

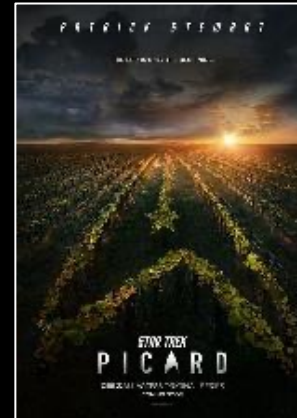
- Machine Learning and AI
- Static vs. Dynamic Typing
- How to estimate Use Cases
- Functional Programming
- Making fun of JavaScript



the 20 year gap

the 2020s is a different era

- Machine Learning and AI
- Static vs. Dynamic Typing
- How to estimate Use Cases
- Functional Programming
- Making fun of JavaScript

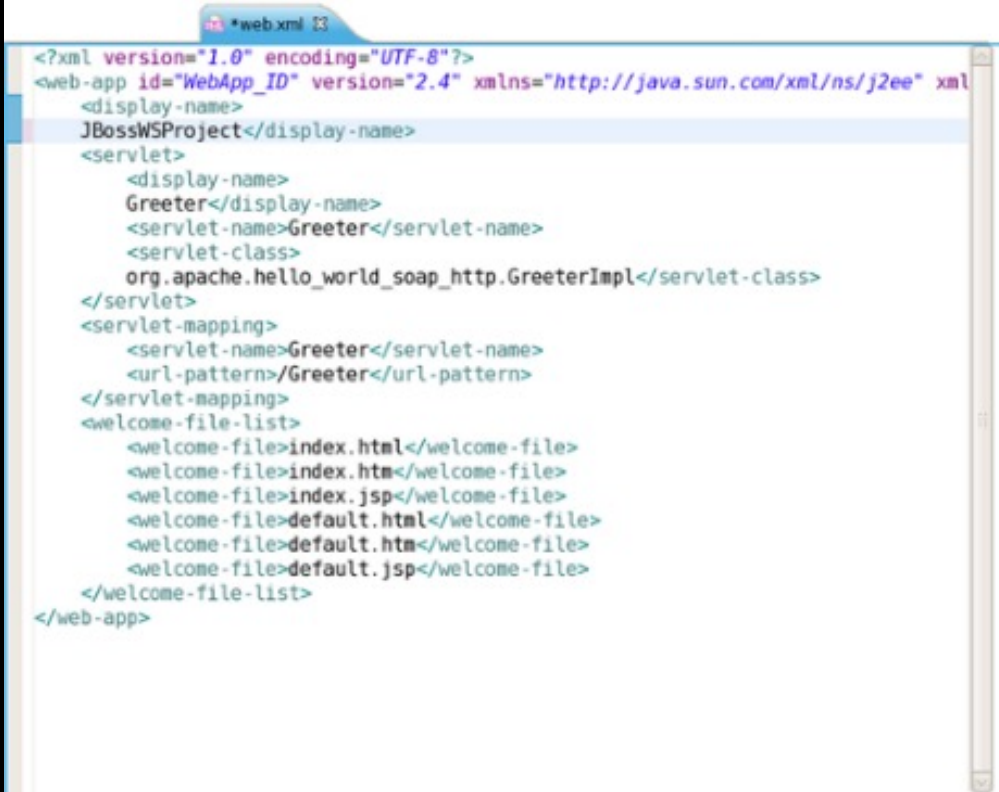


We'd never make that mistake again

the 20 year gap
we had some silly ideas

- **Independent components**
- **Wired together over networks**
- **Convolutd configuration files**

Causing endless confusion with typos and formatting errors

A screenshot of a web.xml file in an IDE. The file is titled '*web.xml' and contains XML code for a web application. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xml
<display-name>
JBossWSProject</display-name>
<servlet>
  <display-name>
    Greeter</display-name>
  <servlet-name>Greeter</servlet-name>
  <servlet-class>
    org.apache.hello_world_soap_http.GreeterImpl</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Greeter</servlet-name>
  <url-pattern>/Greeter</url-pattern>
</servlet-mapping>
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Whoops...

```
ssblog3.yaml x ssblog2.yaml x ssblog4.yaml x
1  AWSTemplateFormatVersion: "2010-09-09"
2  Description: Using existing secure and non-secure SSM Parameters in CloudFormation
3  #
4  # This template creates a MariaDB RDS instance using the following:
5  # A non-secure SSM Parameter for the DB instance class
6  # A secure SSM Parameter for the master password
7  #
8  Resources:
9  MyRDSDB:
10     Type: "AWS::RDS::DBInstance"
11     Properties:
12         # This next line uses a plain-text Parameter Store dynamic reference
13         DBInstanceClass: "{{resolve:ssm:ssbRDSiClass:1}}"
14         AllocatedStorage: '20'
15         Engine: mariadb
16         EngineVersion: '10.2'
17         MasterUsername: appadmin
18         # This next line uses a secure-string Parameter Store dynamic reference
19         MasterUserPassword: "{{resolve:ssm-secure:ssbRDSmEcntl:1}}"
20     Outputs:
21     DbInstanceId:
22         Description: InstanceID of My RDS DB
23         Value: !Ref MyRDSDB
24
```

It is my conjecture that there has been nothing new in computing since the 1970's, with the one exception of AI



@IDOMDAVIS
DOM DAVIS

It is my conjecture that there has been nothing new in computing since the 1970's, with the one exception of AI which was 1984. Everything else is just relearning what we knew already



@IDOMDAVIS
DOM DAVIS

@MARKXA
MARK ALLAN

CLOUD CONSULTANT AND ARCHITECT. AZURE MVP

@JAMIE_ALLEN
JAMIE ALLEN

DIRECTOR AND ARCHITECT. CLOUD, AKKA AND SCALA EXPERT

@JASONBELLDATA
JASON BELL

MACHINE LEARNING EXPERT AND DATA ENGINEER

@JBUNCH77
JAMES BUNCH

SENIOR TECHNICAL ARCHITECT AND MANAGER

@THEDESIGNWRITER
GILLIAN COLHOUN

WRITER WORKING WITH DESIGNERS, BUSINESSES AND UNIVERSITIES

@IDOMDAVIS
DOM DAVIS

CTO, TECHNICAL ARCHITECT AND TECH EVANGELIST

@THETEANERD
DAVID KENNEDY

SOFTWARE ARCHITECT AND LEAD ENGINEER

@JOHNSNOLAN
JOHN NOLAN

TECHNOLOGIST AND ARCHITECT. SPECIALIST IN ETHICS

@OPINIONATEDGEEK
GEOFF TAYLOR

INDEPENDENT SOFTWARE DEVELOPER

@MJPT777
MARTIN THOMPSON

HIGH-PERFORMANCE AND DISTRIBUTED SYSTEMS SPECIALIST

**SOME PREFERRED TO REMAIN ANONYMOUS*

Draw More Pictures

one

draw more pictures

in the 90s we drew design diagrams

- **Then the UML came along and ruined it ☹️**
- **Mostly due to an obsession with 'Executable UML'**

draw more pictures

analysis level diagrams

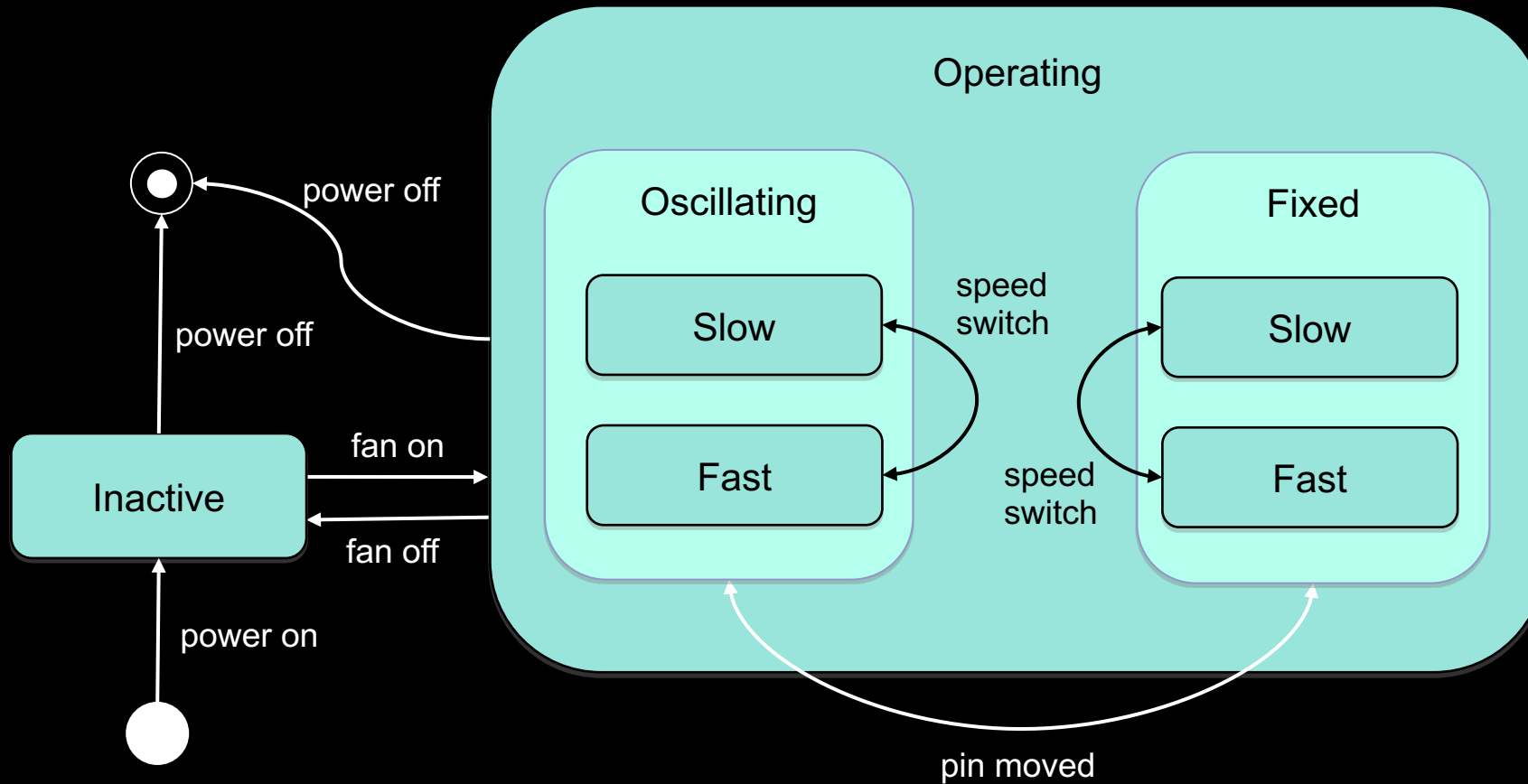


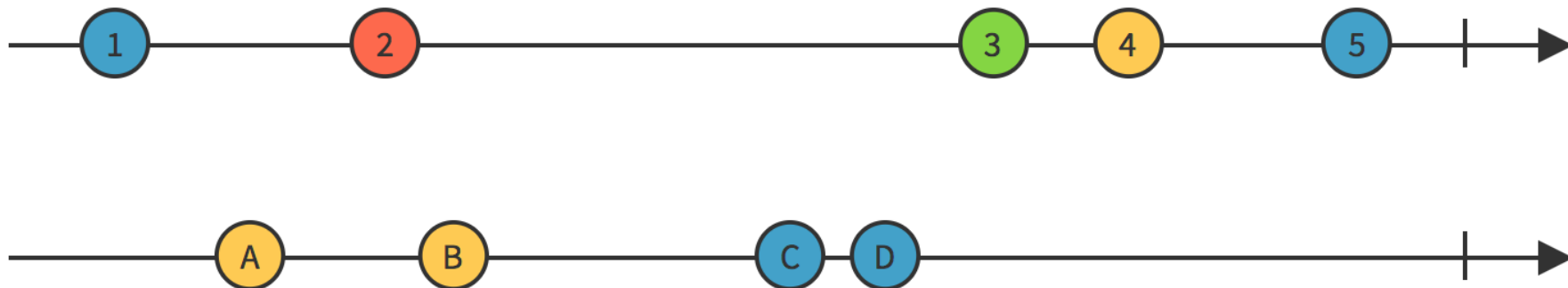
design level diagrams



blueprints of code

draw more pictures





```
combineLatest((x, y) => " " + x + y)
```



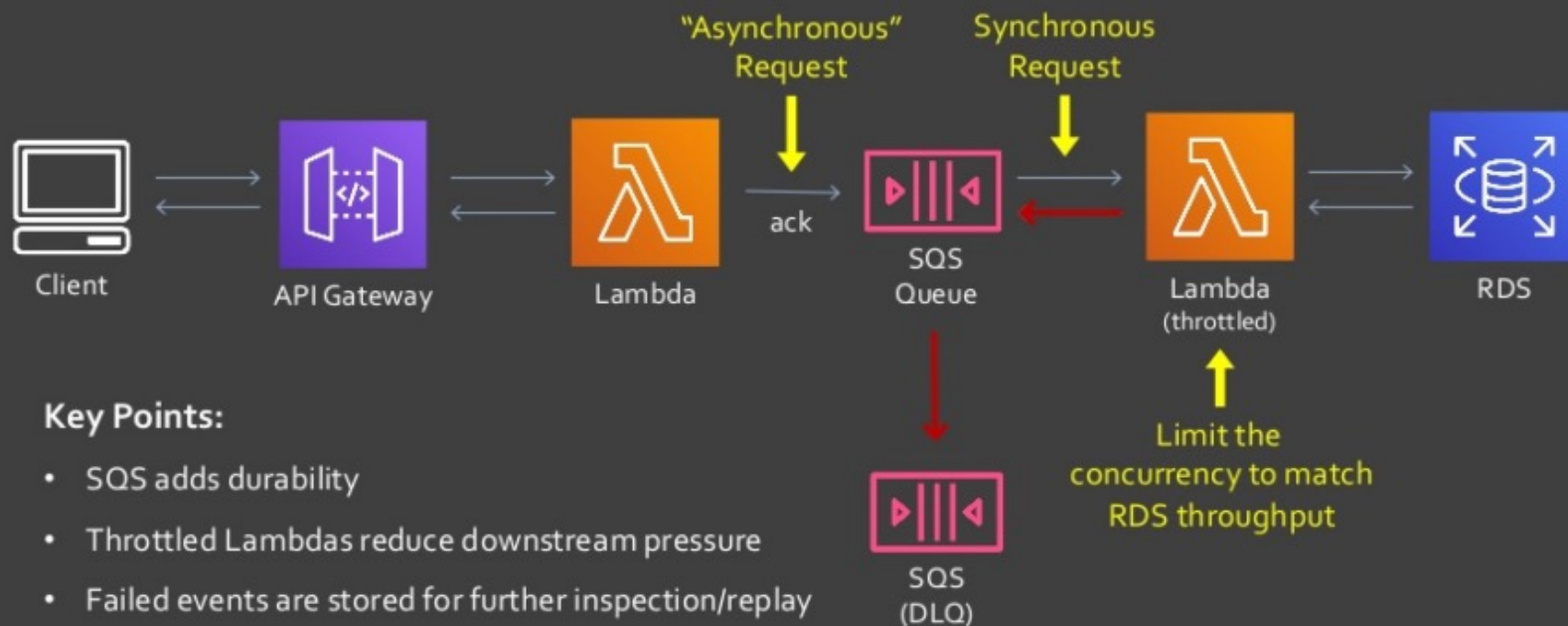
UML was a good idea, fight me. Turning it into a cargo cult of detail was a bad idea.

the people who write wiki articles about the different types of shoes in each Star Trek film got hold of it & ruined it with over-elaboration.



@THETEANERD
DAVID KENNEDY

Buffer events for throttling and durability



Software architects
struggle to communicate
software architecture



9:49 / 48:44



draw more pictures

what can we learn

- **get into the habit of drawing diagrams of architecture**
- **define notations for different domains (mobile, serverless etc.)**
- **create modeling tools that work and don't cost the earth**
- **differentiate between the different levels of abstraction**

Review at the Right Level

two



critique your design

lack of diagrams eroded interest in design

- **In 2000s architecture was commoditized**
 - JEE, RoR etc. – ‘shrink-wrapped design’
- **We stopped doing design reviews**
- **Peer review degenerated into syntax nit-picking**





Einar W. Høst

@einarwh

Replying to [@pati_gallardo](#)

My impression is that we somehow manage to simultaneously undermodel and overengineer.

***Doing a sensible amount up front design
got rejected with a misunderstanding of
agile and dogma taking over.***



@MJPT777

MARTIN THOMPSON



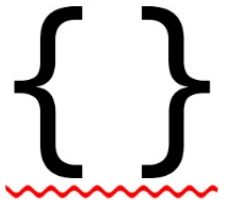
ESLint

sonarqube



StyleCop/**StyleCop**

Analyzes C# source code to enforce a set of style and consistency rules.

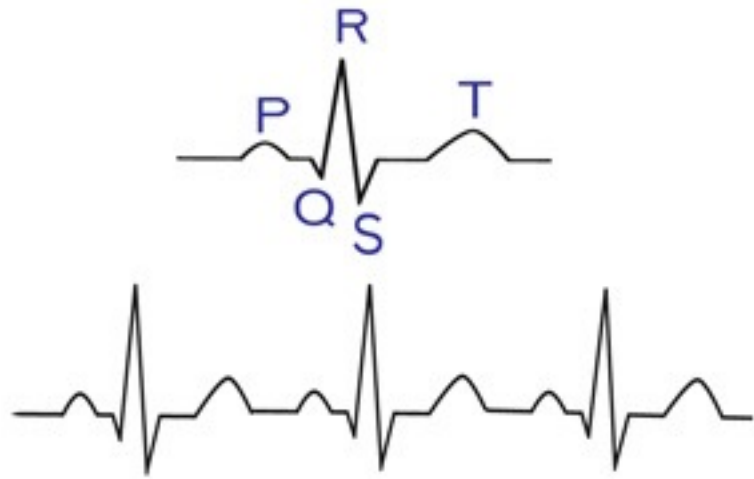


Prettier

ktlint

An anti-bikeshedding Kotlin linter with built-in formatter.

Normal heart rhythm



Irregular heart rhythm



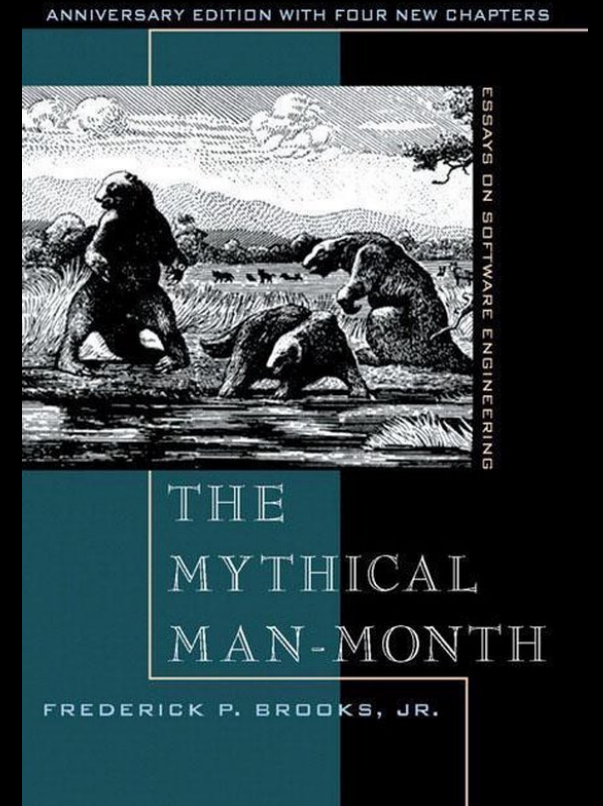
review at the right level

what can we learn

- make design reviews part of our process**
- run the code we are supposed to be reviewing**
- look for and promote the use of patterns in code**
- step back from syntax and focus on algorithms**

Value Stability

three



“There’s no such thing as
advanced developers”

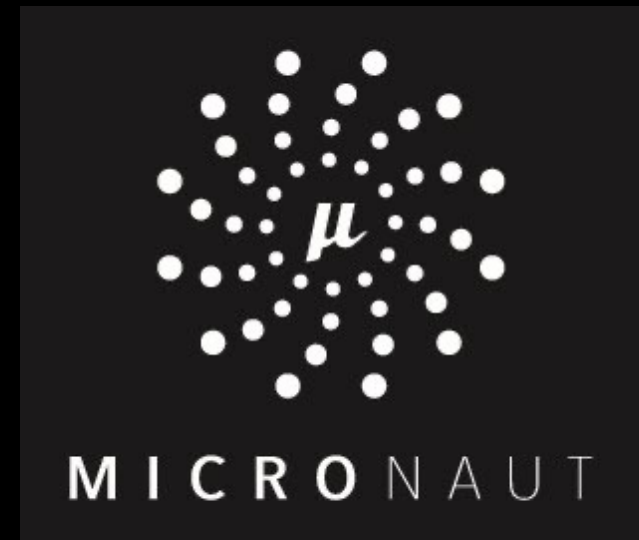
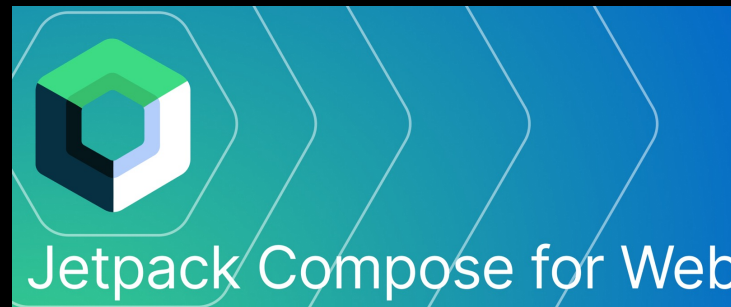
we are surrounded by choices
...and the choices change



thought experiment

could you...

- **Name five JVM microservice frameworks?**
- **Name five libraries for managing state in React?**
- **Name five mechanisms for deploying to AWS?**





Redux



Mobx

PULLSTATE



Redux Toolkit



Relay



Recoil

1. Web Console
2. AWS CLI
3. Cloud Formation Template
 - By CDK
 - By SAM
 - Via pure CF JSON
 - Via YAML
4. Writing an Ansible playbook
5. Writing Terraform
6. Chef Automate

7. Puppet Enterprise
8. AWS Elastic Beanstalk
9. AWS CodeDeploy
10. AWS CodeStar
11. AWS OpsWorks
12. Cloud Foundry Bosh

Everything
begins with
choice



there is increased demand for Advocates plus Heroes, Champions etc...

Sy Brand

@TartanLlama

c++ developer advocate @microsoft ~ poet ~ filmmaker ~ tired person ~ they/them

Pratim

@BhosalePratim

Developer Advocate @nhostio | Helping Devs scale their Backend | ex VP Growth @ShowcaseHQ | Web3 enthusiast | #SpacesHost

Josh Long

@starbuxman

Spring Developer Advocate (JC & Kotlin GDE) @VMwareTanzu Check out joshlong.com for international names, books, podcast, screencasts, & Livelessons

Hadi Hariri

@hhariri Follows you

Developer Advocacy at @JetBrains.

Eric Johnson

@edjgeek

Principal Developer Advocate @AWScloud. speaker - serverless & overcoming challenges. Opinions are my own. Do justice, love mercy, walk humbly. New Grandpa!!

Pati

@patischerbatsky

Developer Relations @Google, @GoogleDevEurope

José Paumard

@JosePaumard Follows you

#Java #DevRel #Advocate @Oracle, bartender #JEPCafe and dev.java, Java Champion Alumni, JavaOne Rockstar, Java enthusiast. Opinions are my own

Sebastian Aigner

@sebi_io Follows you

Dev Advocate @JetBrains. Eternally learning & teaching Kotlin. Hosting @talkingkotlin. Once you've worked with love, you won't want to work in any other way.

Martin Woodward

@martinwoodward

Senior Director of DevRel for @GitHub. Previously Executive Director @dotnetfdn and original creator of the @Microsoft org on @GitHub (he/him)

Nicolai Parlog

@nipafx

#Java enthusiast with a passion for learning and sharing - in posts, newsletters, videos, streams, talks... Developer Advocate at Oracle, organizer of @AccentoDev

Mark Heckler, 馬克 哈格樂, مارک هکالر

@mkheck

Dev/Advocate @microsoft, comp scientist+MBA, pilot, inglés+español. @Java_Champions, @GoogleDevExpert in @Kotlin, Rockstar presenter, author @springbootbook

Timur Doumler

@timur_audio

Co-founder @cradleaudio. Developer Advocate @jetbrains. Organiser @include_cpp. Conference speaker. ISO C++ committee member. Pastafarian. Socialist. he/him

“Developers don’t read books”

“Blogs shouldn’t be wordy”

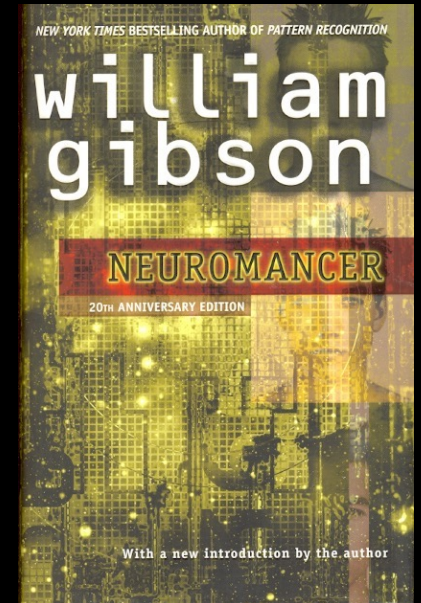
“What’s a specification?”

read and write

it was foreseen we would become illiterate

Minds aren't read. See, you've still got the paradigms print gave you, and you're barely print literate.

Gibson, William





Donna Amos



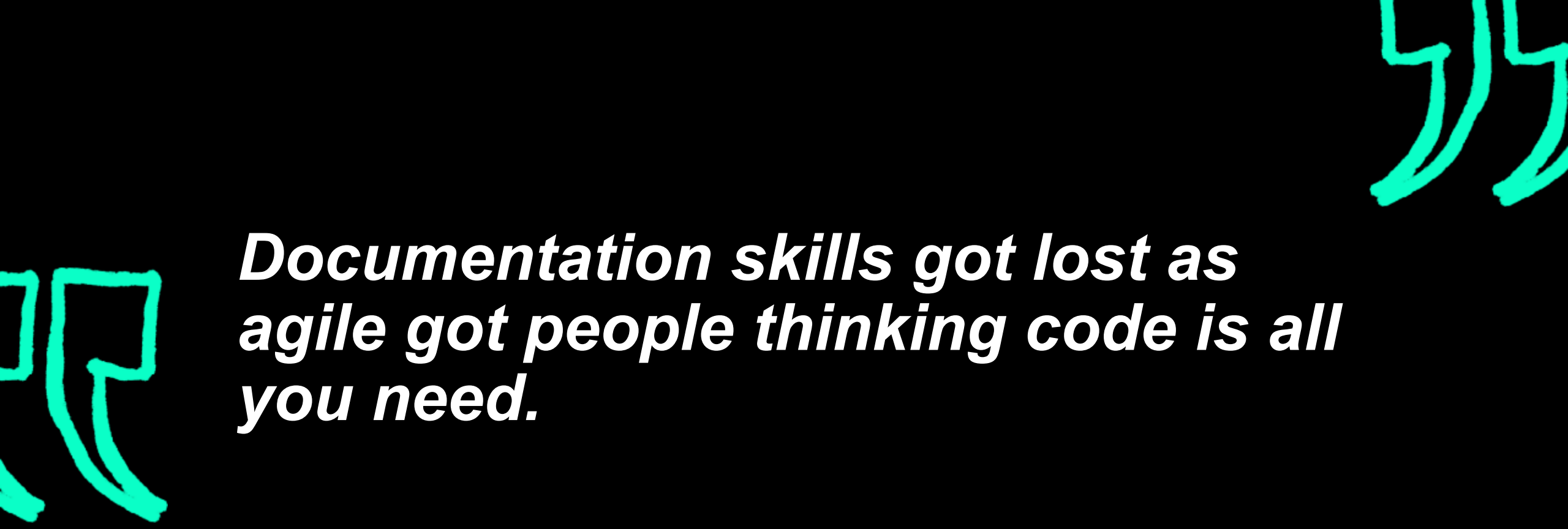
@donnacamos88



Replying to [@babiescatscode](#)

If you can understand it without a Youtube video and 5 separate blog posts, is it really documentation though?

7:08 PM · Jan 20, 2020 · [Twitter Web App](#)



Documentation skills got lost as agile got people thinking code is all you need.



@MJPT777

MARTIN THOMPSON

Specifications are a good thing

 LIVE

[#2022FRENCH ELECTIONS](#) [#WAR IN UKRAINE](#) [FRANCE](#) [AFRICA](#) [CULTURE](#) [SHOWS](#) [FIGHT THE FAKE](#)

The EU member states and parliament on Tuesday agreed the text of a law imposing a standard charger for smartphones and tablets sold in the bloc, in a blow to Apple. For most portable devices the requirement for charging via a USB Type-C port will come into effect from late 2024, the European Parliament's press service said.

The [EU](#) believes a standard cable for all devices will cut back on electronic waste, but iPhone juggernaut Apple argues that a one-size-fits-all charger would slow innovation and create more [pollution](#).

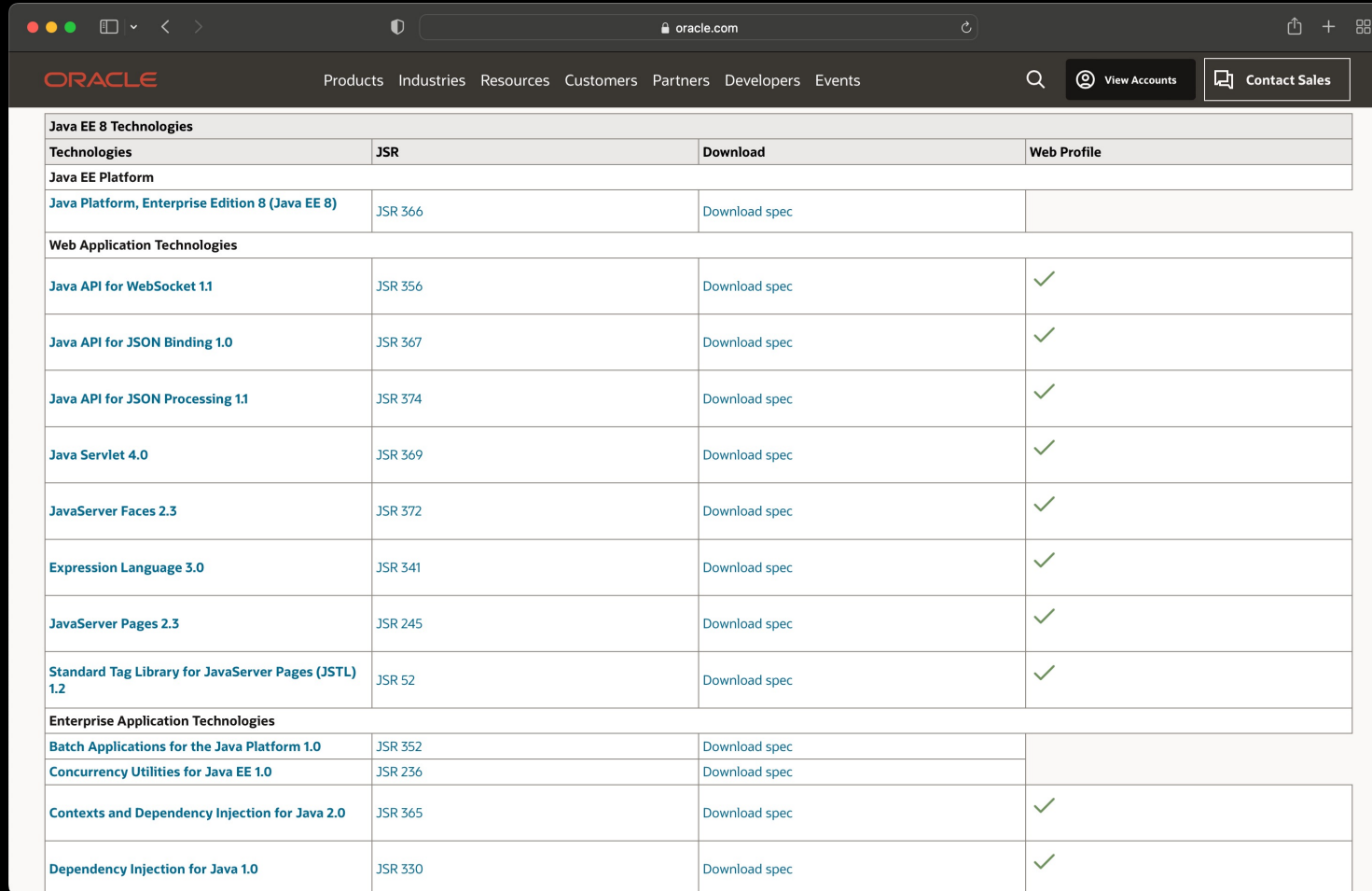
The bloc is home to 450 million people, some of the world's richest consumers, and the imposition of the USB-C as a cable standard could affect the entire global [smartphone](#) market.

"This new law will make the lives of European consumers easier and will be better for the environment," said MEP Andrey Kovatchev, one of the negotiators.

"It is time to put an end to the piles of cables that we all have in our drawers and save about 11,000 tonnes of electronic waste per year," he said.



Specifications keep vendors honest!

A screenshot of a web browser displaying the Oracle Java EE 8 Technologies page. The browser's address bar shows 'oracle.com'. The page has a dark header with the Oracle logo and navigation links: Products, Industries, Resources, Customers, Partners, Developers, and Events. On the right of the header are search and account links. The main content is a table with four columns: Technologies, JSR, Download, and Web Profile. The table is organized into sections: Java EE 8 Technologies, Web Application Technologies, and Enterprise Application Technologies. The 'Web Profile' column contains green checkmarks for most technologies, indicating they are supported in a specific web profile.

Java EE 8 Technologies			
Technologies	JSR	Download	Web Profile
Java EE Platform			
Java Platform, Enterprise Edition 8 (Java EE 8)	JSR 366	Download spec	
Web Application Technologies			
Java API for WebSocket 1.1	JSR 356	Download spec	✓
Java API for JSON Binding 1.0	JSR 367	Download spec	✓
Java API for JSON Processing 1.1	JSR 374	Download spec	✓
Java Servlet 4.0	JSR 369	Download spec	✓
JavaServer Faces 2.3	JSR 372	Download spec	✓
Expression Language 3.0	JSR 341	Download spec	✓
JavaServer Pages 2.3	JSR 245	Download spec	✓
Standard Tag Library for JavaServer Pages (JSTL) 1.2	JSR 52	Download spec	✓
Enterprise Application Technologies			
Batch Applications for the Java Platform 1.0	JSR 352	Download spec	
Concurrency Utilities for Java EE 1.0	JSR 236	Download spec	
Contexts and Dependency Injection for Java 2.0	JSR 365	Download spec	✓
Dependency Injection for Java 1.0	JSR 330	Download spec	✓

value stability

what can we learn

- **break the cycle of badly written documentation**
- **educate ourselves to write quality documentation**
- **resist the temptation to 'wait for the next big thing'**
- **take a deep interest into how frameworks work**
- **push vendors to create and comply to specifications**

Invent It Here

four



invent it here

in the 90s software companies wrote lots
of their own libraries and tools





The NIH syndrome (Not Invented Here) is a disease.

— *Linus Torvalds* —

AZ QUOTES

invent it here

but today we've swung to the other extreme

- **Often these days we don't know what we are using**
- **Projects typically have huge trees of dependencies**
- **We cross our fingers and hope this won't be a problem**

We always go looking for a library-based solution

```
$ create-react-app --typescript simple-form
```

```
...
```

```
[1/4] Resolving packages...
```

```
[2/4] Fetching packages...
```

```
info fsevents@1.2.11: The platform "win32" is incompatible with this module.
```

```
info "fsevents@1.2.11" is an optional dependency and failed compatibility check. Excluding it from installation.
```

```
info fsevents@2.1.2: The platform "win32" is incompatible with this module.
```

```
info "fsevents@2.1.2" is an optional dependency and failed compatibility check. Excluding it from installation.
```

```
[3/4] Linking dependencies...
```

```
warning "react-scripts > eslint-config-react-app@5.2.0" has incorrect peer dependency "eslint-plugin-flowtype@3.x".
```

```
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.17.1" has unmet peer dependency
```

```
"typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta".
```

```
$ cd simple-form
```

```
$ yarn list --depth 0 | wc -l
```

```
1190
```



Brad Frost
@brad_frost

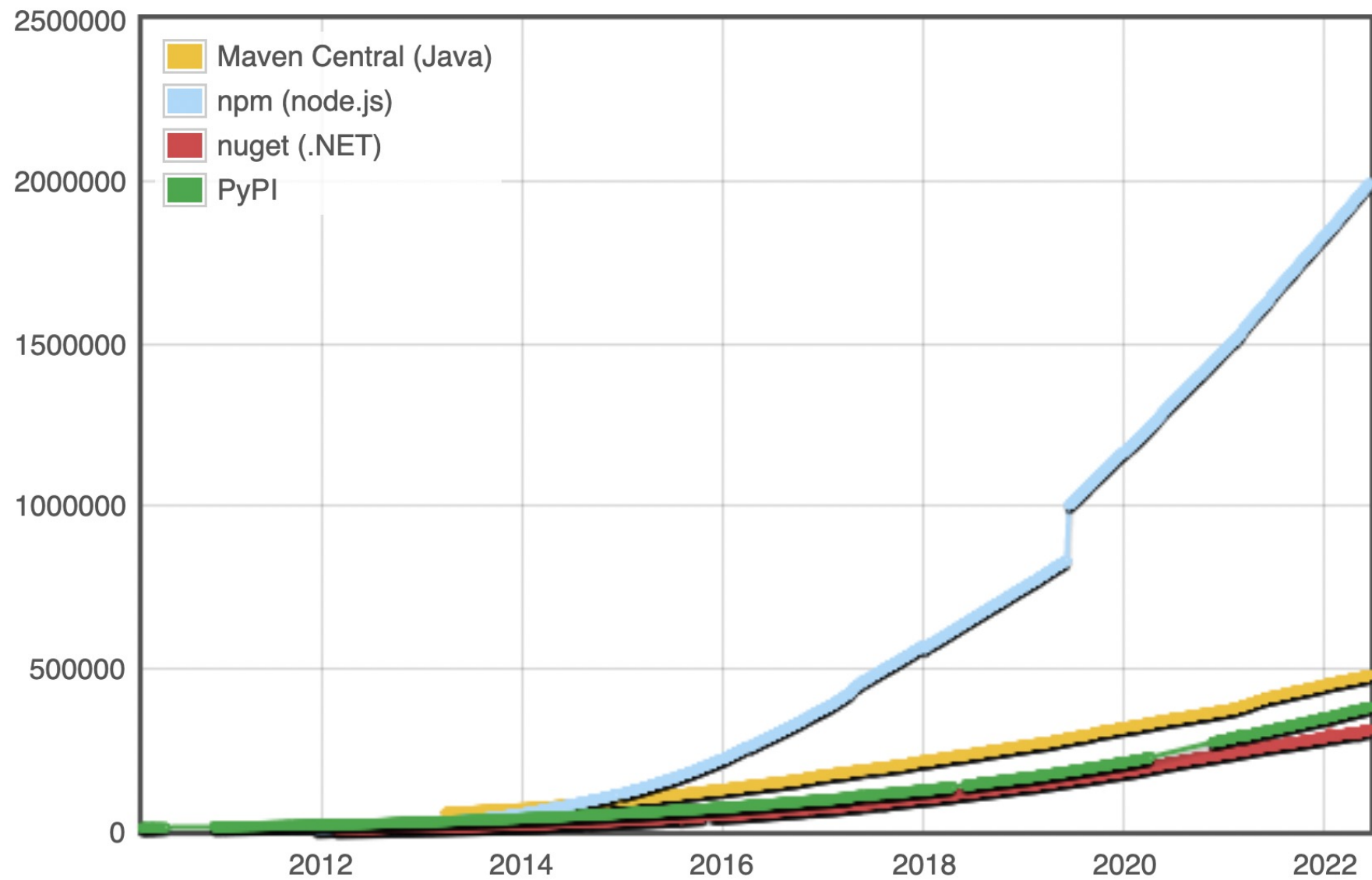
npm install



2:08 PM · May 14, 2018 · Twitter for Android



<https://turnoff.us/geek/npm-install/>



<http://www.modulecounts.com/>



Developer loved, Security trusted.

Find and automatically fix vulnerabilities in your code, open source dependencies, containers, and infrastructure as code — all powered by Snyk's industry-leading security intelligence.

Get started with a free forever account, and scale up if needed.



```
→ snyk test
Testing /Users/gooddog/projects/woof...
Tested 3 dependencies for known issues, found 2 issues, 2 vulnerable paths.

x Remote Code Execution (RCE) [Critical Severity] in log4j-core@2.14.1
  introduced by io.gooddog:woof@0.0.1
    This issue was fixed in versions: 2.3.1, 2.12.2, 2.15.0

x Remote Code Execution (RCE) [Critical Severity] in log4j-core@2.14.1
  introduced by io.gooddog:woof@0.0.1
    This issue was fixed in versions: 2.3.1, 2.12.2, 2.16.0

Package manager:  maven
Target file:      pom.xml
Project name:     io.gooddog:woof
```

frameworks == remote development



Kelsey Hightower 

@kelseyhightower

Remote dev environments are trending. Seems the process of writing software has become so complex that 10 cores and 64 GB of RAM isn't enough.

Or maybe this has more to do with the growing number of external dependencies and the related configuration required to manage it all.

3:26 · 09 Jun 22 · [Twitter Web App](#)

***Packaging in C was hard, but it did
make you intensely aware of your
dependencies!***



@OPINIONATEDGEEK
GEOFF TAYLOR

invent it here

what can we learn

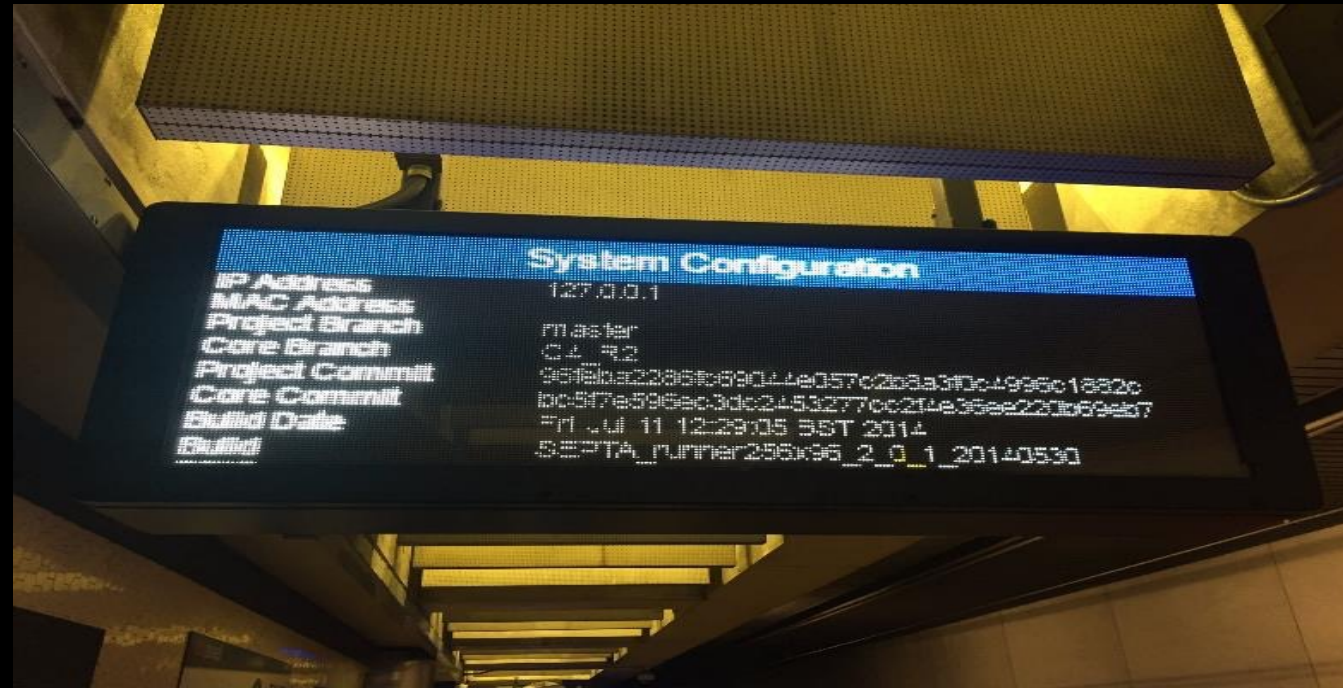
- **take control of the dependencies of your applications**
- **evaluate the cost/benefit of using external libraries**

Learn to Test

five

learn to test

my favourite #kevinhenney



Boeing's 2nd Starliner software glitch could have led to an in-space collision

By [Hanneke Weitering](#) 8 hours ago

Starliner's crew module could have "bumped" into its service module after separation if Boeing hadn't caught the error.



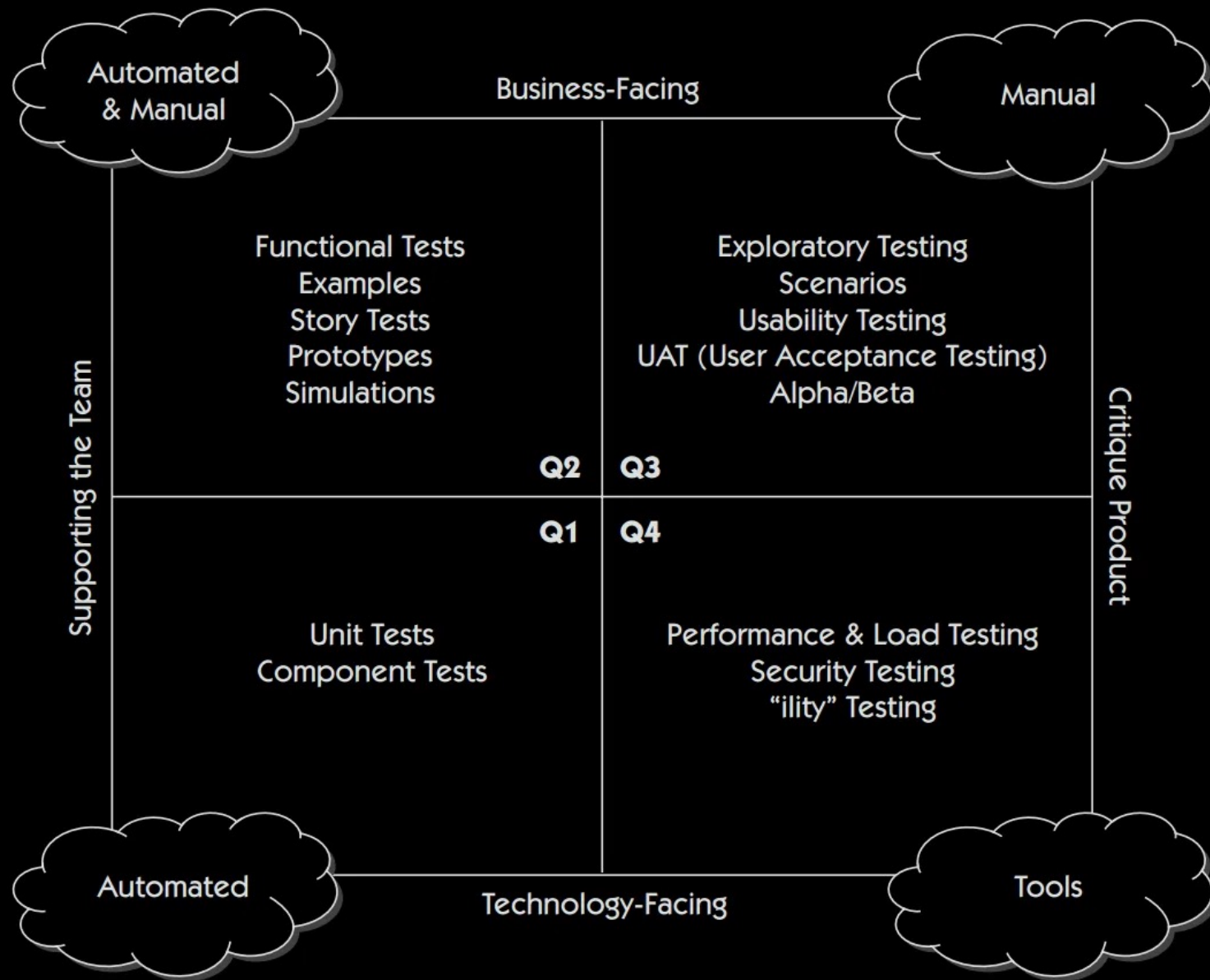
Boeing's CST-100 Starliner spacecraft is pictured back home at the company's Commercial Crew and Cargo Processing Facility, where it is undergoing inspection after its Orbital Flight Test mission in December 2019. (Image credit: Frank Michaux/NASA)

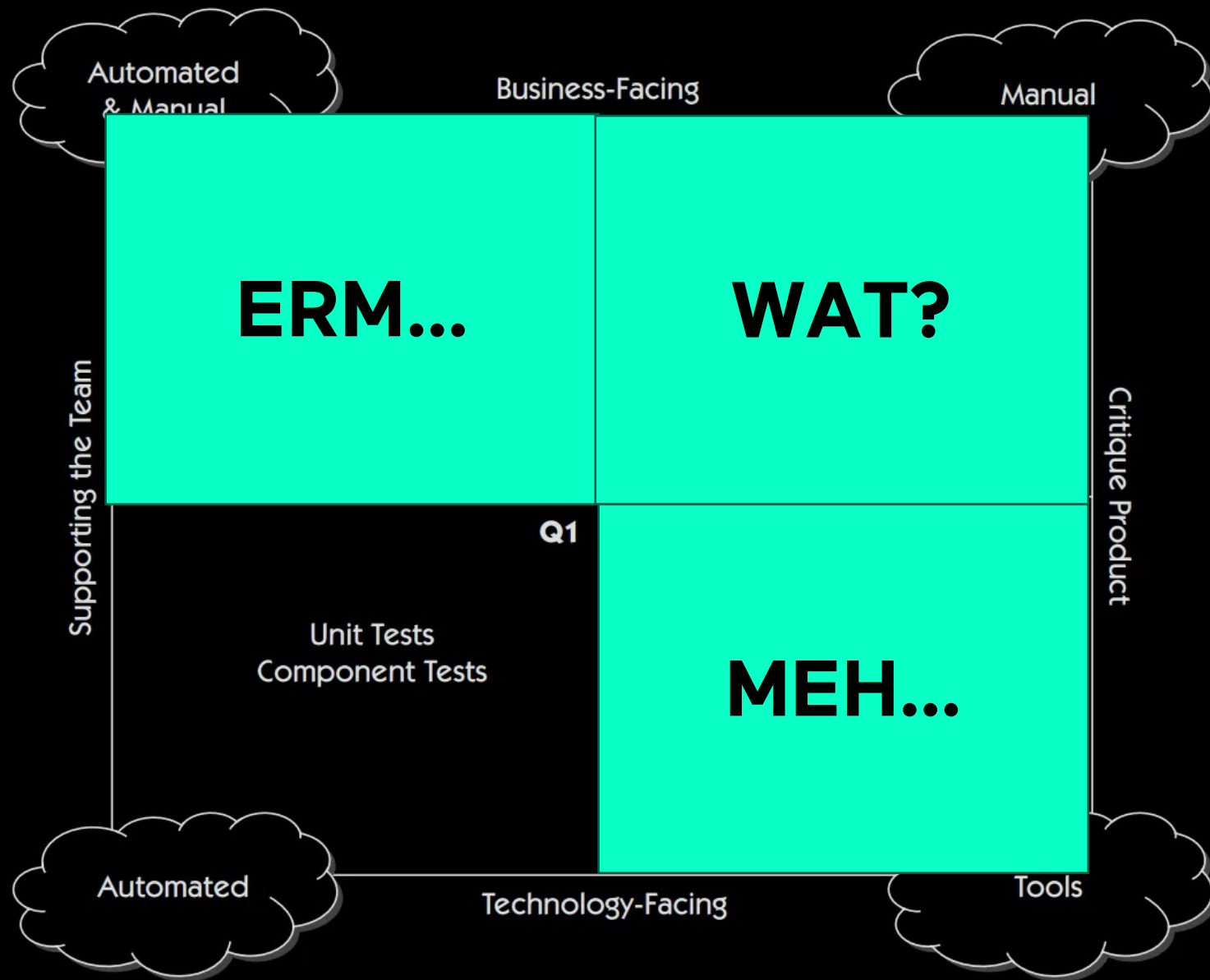
Seven weeks after [Boeing's CST-100 Starliner](#) failed to reach the International Space Station as planned during its first orbital flight test, NASA and Boeing officials disclosed the preliminary results of an investigation into what went wrong.

learn to test

agile championed developers writing tests

- **But it didn't always instill respect for testing**
- **Developers today seem to have forgotten**
 - There is more to testing than unit tests
 - Not all forms of testing can be automated (easily)
- **The profession of testing has been evolving**







MODERN TESTING PRINCIPLES

BY ALAN PAGE AND BRENT JENSEN

learn to test

what can we learn

- **have more respect for the tester's role**
- **understand the whole scope of testing**
- **realize not all testing can be automated**
- **take non-functional testing seriously**



usability
security
accessibility
performance
internationalization

Master the Tools

six

master the tools

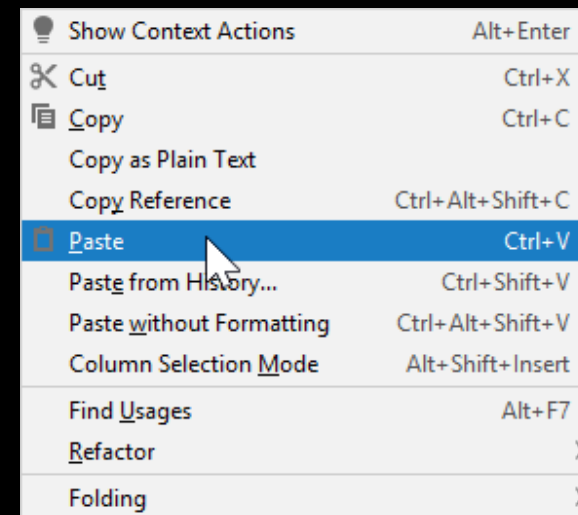
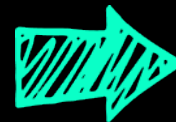
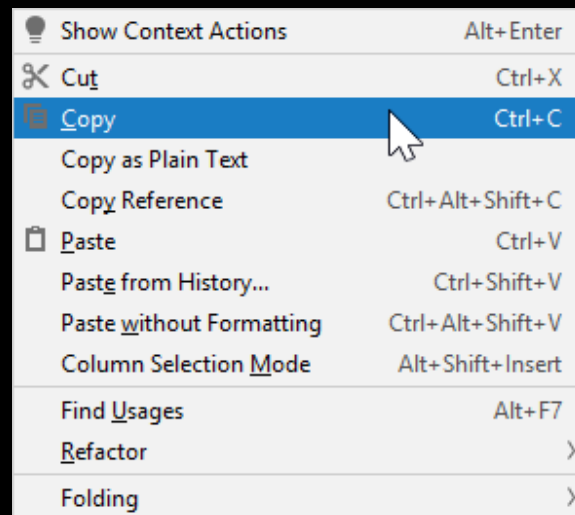
optimise your workflows

- **Longer compiles and slow machines forced us to streamline**
 - Keyboard Shortcuts
 - Macros
 - Multi-task

master the tools

we've never had so much opportunity

- **Yet, many people don't leverage this**
 - Afraid of the terminal
 - One hand never leaves the mouse



VIM - Vi IMproved
version 8.1.22
by Bram Moolenaar et al.
vim is open source and freely distributable

Help poor children in Uganda!
type :help iccf<Enter> for information

type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info



DEFAULT KEYMAP



Remember these Shortcuts

Smart code completion	Ctrl + Shift + Space
Search everywhere	Double Shift
Show intention actions and quick-fixes	Alt + Enter
Generate code	Alt + Ins
Parameter info	Ctrl + P
Extend selection	Ctrl + W
Shrink selection	Ctrl + Shift + W
Recent files popup	Ctrl + E
Rename	Shift + F6

General

Open corresponding tool window	Alt + #[0-9]
Save all	Ctrl + S
Synchronize	Ctrl + Alt + Y
Toggle maximizing editor	Ctrl + Shift + F12
Inspect current file with current profile	Alt + Shift + I
Quick switch current scheme	Ctrl + BackQuote (`)
Open Settings dialog	Ctrl + Alt + S
Open Project Structure dialog	Ctrl + Alt + Shift + S
Find Action	Ctrl + Shift + A

Debugging

Step over / into	F8 / F7
Smart step into / Step out	Shift + F7 / Shift + F8
Run to cursor	Alt + F9
Evaluate expression	Alt + F8
Resume program	F9
Toggle breakpoint	Ctrl + F8
View breakpoints	Ctrl + Shift + F8

Search / Replace

Search everywhere	Double Shift
Find	Ctrl + F
Find next / previous	F3 / Shift + F3
Replace	Ctrl + R
Find in path	Ctrl + Shift + F
Replace in path	Ctrl + Shift + R
Select next occurrence	Alt + J
Select all occurrences	Ctrl + Alt + Shift + J
Unselect occurrence	Alt + Shift + J

—Productivity Boosters

Editing

Basic code completion	Ctrl + Space
Smart code completion	Ctrl + Shift + Space
Complete statement	Ctrl + Shift + Enter
Parameter info (within method call arguments)	Ctrl + P
Quick documentation lookup	Ctrl + Q
External Doc	Shift + F1
Brief Info	Ctrl + mouse
Show descriptions of error at caret	Ctrl + F1
Generate code...	Alt + Insert
Override methods	Ctrl + O
Implement methods	Ctrl + I
Surround with...	Ctrl + Alt + T
Comment / uncomment with line comment	Ctrl + /
Comment / uncomment with block comment	Ctrl + Shift + /
Extend selection	Ctrl + W
Shrink selection	Ctrl + Shift + W
Context info	Alt + Q
Show intention actions and quick-fixes	Alt + Enter
Reformat code	Ctrl + Alt + L
Optimize imports	Ctrl + Alt + O
Auto-indent line(s)	Ctrl + Alt + I
Indent / unindent selected lines	Tab / Shift + Tab
Cut current line to clipboard	Ctrl + X, Shift + Delete
Copy current line to clipboard	Ctrl + C, Ctrl + Insert
Paste from clipboard	Ctrl + V, Shift + Insert
Paste from recent buffers...	Ctrl + Shift + V
Duplicate current line	Ctrl + D
Delete line at caret	Ctrl + Y
Smart line join	Ctrl + Shift + J
Smart line split	Ctrl + Enter
Start new line	Shift + Enter
Toggle case for word at caret or selected block	Ctrl + Shift + U
Select till code block end / start	Ctrl + Shift +] / [
Delete to word end	Ctrl + Delete
Delete to word start	Ctrl + Backspace
Expand / collapse code block	Ctrl + NumPad+ / -
Expand all	Ctrl + Shift + NumPad+
Collapse all	Ctrl + Shift + NumPad-
Close active editor tab	Ctrl + F4

Refactoring

Copy	F5
Move	F6
Safe Delete	Alt + Delete
Rename	Shift + F6
Refactor this	Ctrl + Alt + Shift + T
Change Signature	Ctrl + F6
Inline	Ctrl + Alt + N
Extract Method	Ctrl + Alt + M
Extract Variable	Ctrl + Alt + V
Extract Field	Ctrl + Alt + F
Extract Constant	Ctrl + Alt + C
Extract Parameter	Ctrl + Alt + P

Navigation

Go to class	Ctrl + N
Go to file	Ctrl + Shift + N
Go to symbol	Ctrl + Alt + Shift + N
Go to next / previous editor tab	Alt + Right/Left
Go back to previous tool window	F12
Go to editor (from tool window)	Esc
Hide active or last active window	Shift + Esc
Go to line	Ctrl + G
Recent files popup	Ctrl + E
Navigate back / forward	Ctrl + Alt + Left/Right
Navigate to last edit location	Ctrl + Shift + Backspace
Select current file or symbol in any view	Alt + F1
Go to declaration	Ctrl + B, Ctrl + Click
Go to implementation(s)	Ctrl + Alt + B
Open quick definition lookup	Ctrl + Shift + I
Go to type declaration	Ctrl + Shift + B
Go to super-method / super-class	Ctrl + U
Go to previous / next method	Alt + Up/Down
Move to code block end / start	Ctrl +]/[
File structure popup	Ctrl + F12
Type hierarchy	Ctrl + H
Method hierarchy	Ctrl + Shift + H
Call hierarchy	Ctrl + Alt + H
Next / previous highlighted error	F2 / Shift + F2
Edit source / View source	F4 / Ctrl + Enter
Show navigation bar	Alt + Home
Toggle bookmark	F11
Toggle bookmark with mnemonic	Ctrl + F11
Go to numbered bookmark	Ctrl + #[0-9]
Show bookmarks	Shift + F11

Compile and Run

Make project	Ctrl + F9
Compile selected file, package or module	Ctrl + Shift + F9
Select configuration and run / debug	Alt + Shift + F10/F9
Run / Debug	Shift + F10 / F9
Run context configuration from editor	Ctrl + Shift + F10

Usage Search

Find usages / Find usages in file	Alt + F7 / Ctrl + F7
Highlight usages in file	Ctrl + Shift + F7
Show usages	Ctrl + Alt + F7

VCS / Local History

Commit project to VCS	Ctrl + K
Update project from VCS	Ctrl + T
Push commits	Ctrl + Shift + K
'VCS' quick popup	Alt + BackQuote (`)

Live Templates

Surround with Live Template	Ctrl + Alt + J
Insert Live Template	Ctrl + J

master the tools

what can we learn

- learn keyboard shortcuts**
- master your IDE**
- be comfortable with the terminal**
- try to be “lazy”**
- automate the boring stuff**

Focus on Fundamentals

seven

focus on fundamentals

we've taught over a thousand courses

- **We see less focus on fundamentals**
 - How to maximize / measure performance
 - Data structures and algorithms
 - What alternative collection libraries exist
 - Details of compiler optimization and GC
 - Low level info on threading and networks

The ability to design for asynchronous systems got lost as did wonders of state machines. People who worked on games, complex UIs, or device drivers had no issue with asynchronous designs.



@MJPT777

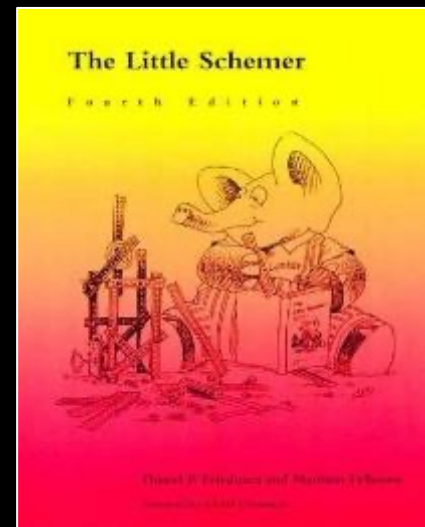
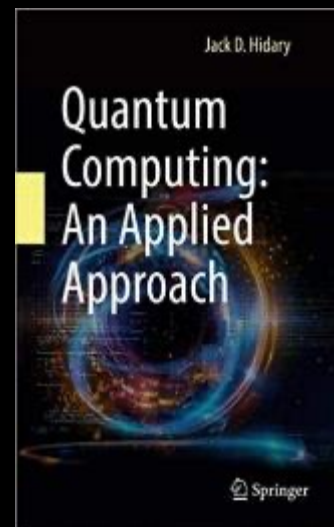
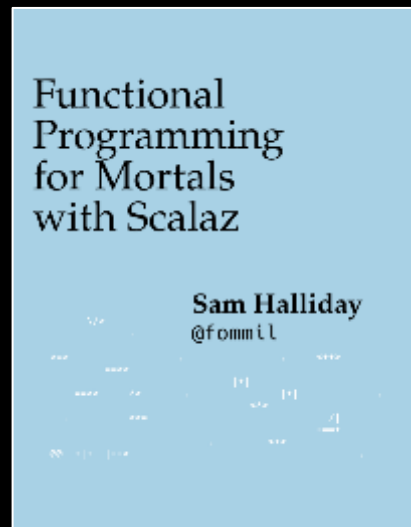
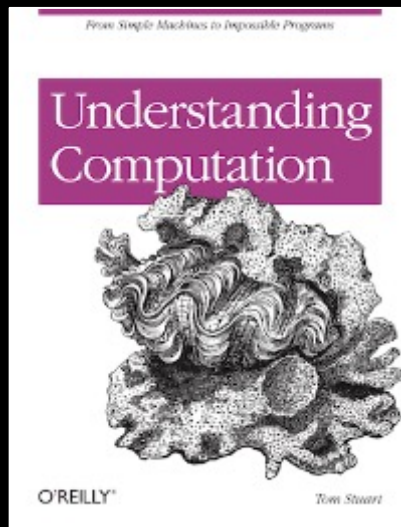
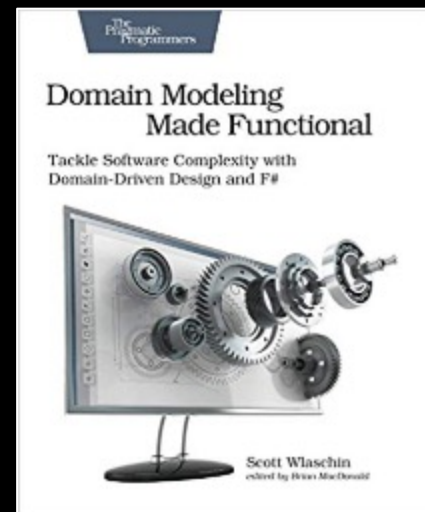
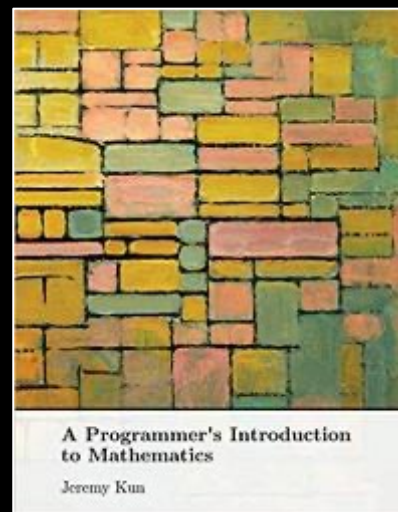
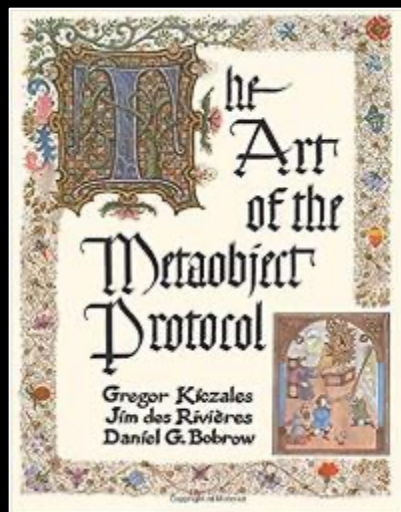
MARTIN THOMPSON

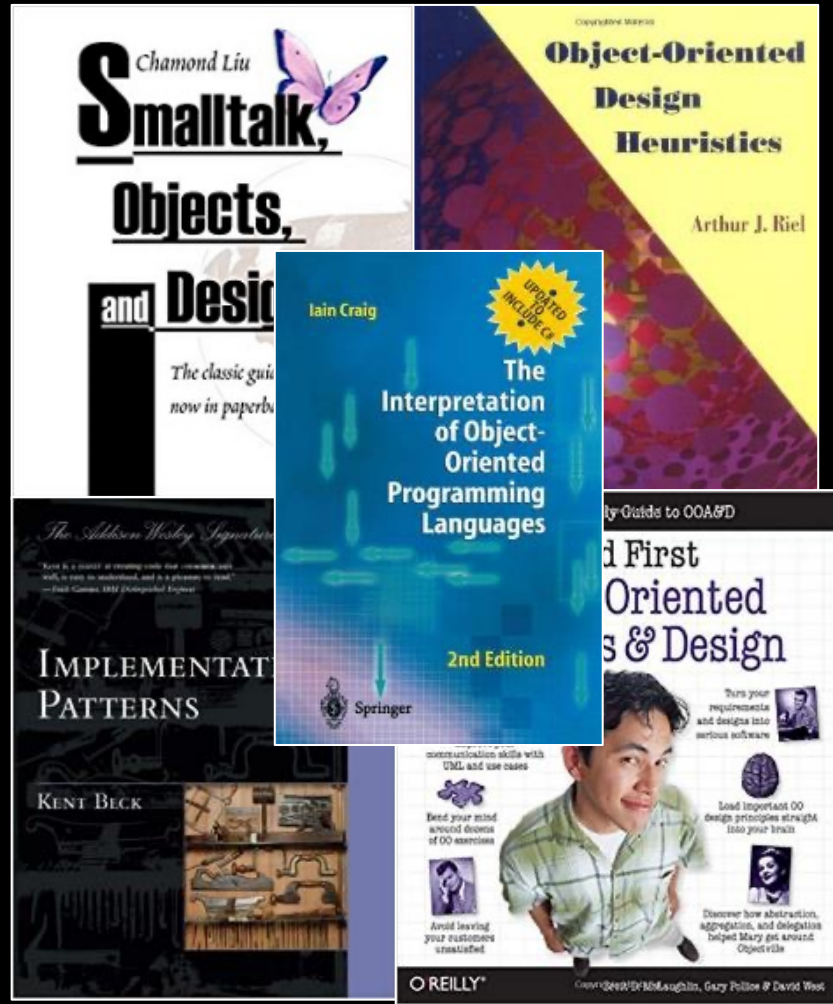
focus on fundamentals

take the 'Bruce Wayne approach'

- **Focus on 'Root Styles' not 'Shiny Things'**
 - Procedural (Pull Based)
 - Declarative (Push Based)
 - Functional (Lambda Calculus)
 - Object Oriented (Behavior)
 - Generic (Type Based)







Back in the days when programmers were women and hardware was expensive, we could [do amazing things].

If we were still coding with that care and ingenuity who knows what we could achieve? At the very least we'd be making a sizeable dent in the climate emergency.



@MARKXA

MARK ALLAN

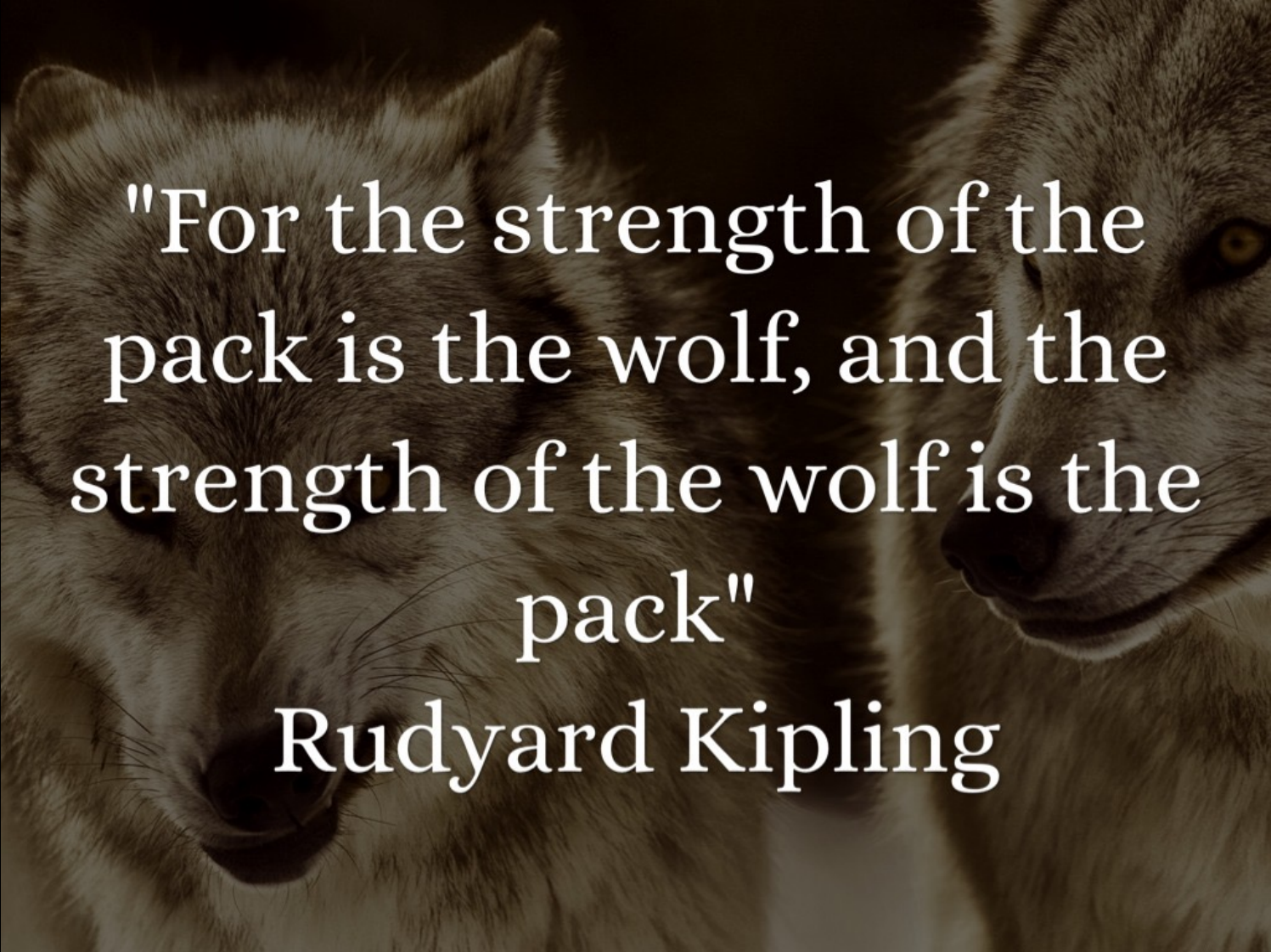
focus on fundamentals

what can we learn

- **focus on re-useable skills**
- **algorithms, data structures, clean code, design principles etc.**
- **avoid spending time (re) learning the next shiny framework**

**remain
accountable**

eight

A close-up photograph of two wolves. The wolf on the left is looking down, while the wolf on the right is looking towards the camera. The image is dark and moody, with the wolves' fur appearing in shades of grey and brown. The text is overlaid in a white, serif font.

"For the strength of the
pack is the wolf, and the
strength of the wolf is the
pack"

Rudyard Kipling

remain accountable

agile was all about professionals

- **Coming together in self-organizing teams**
- **Not blindly following process regardless of outcomes**
- **As with technical skills, a professional outlook was assumed**
- **The focus on the group was not meant to excuse the individual**



Be one of many hungry hunters in a pack, not
one of many docile sheep in a herd



jason gorman
@jason gorman



Agility is when your employees get to act like grown-ups

11:59 PM · Feb 7, 2020 · [Twitter Web App](#)

remain accountable

what can we learn

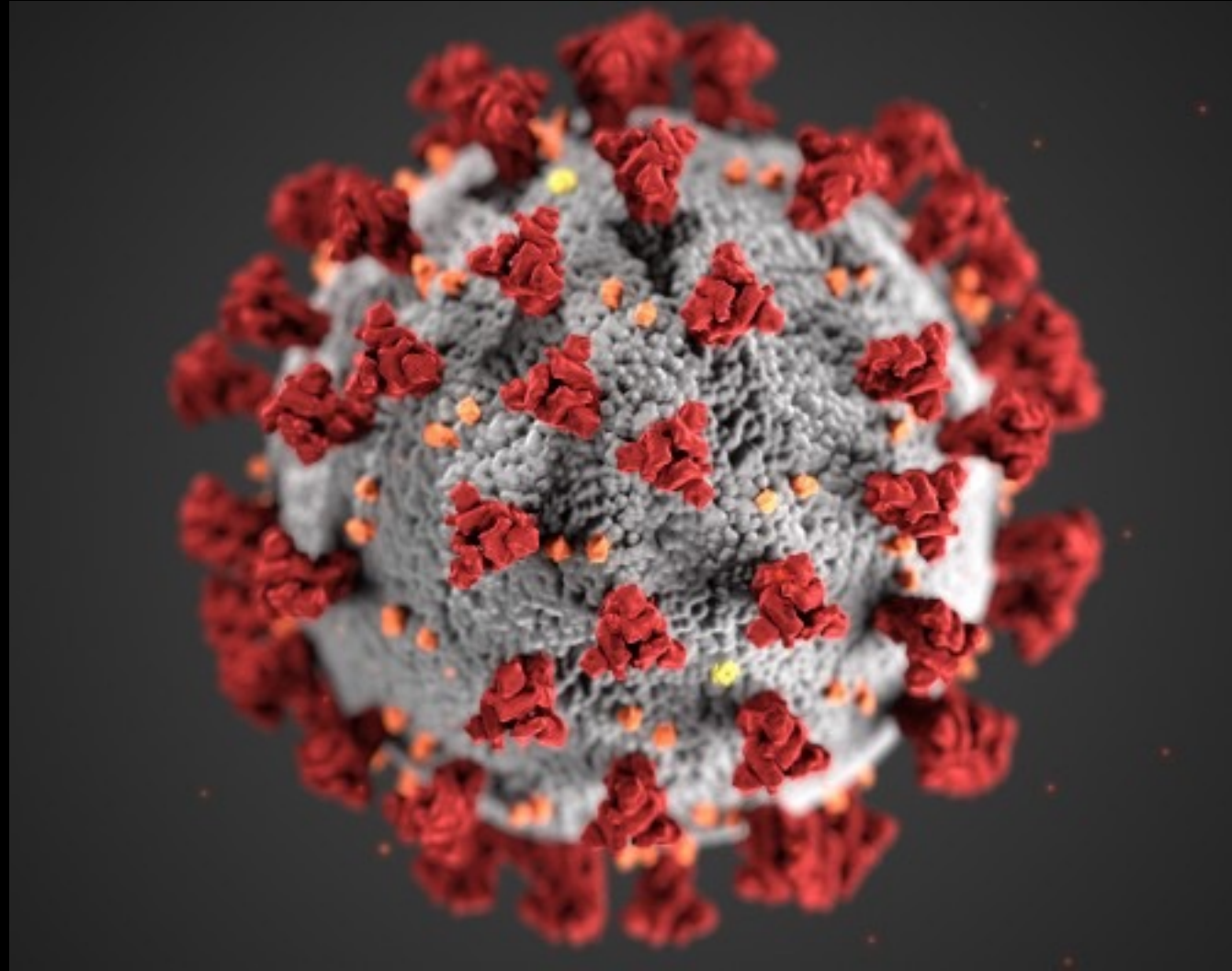
- **take responsibility and be accountable for your work**
- **focus principles over process – understand the why**
- **be a professional**

Prepare for Rain

nine



It's not always possible to see what's around the corner



prepare for rain

many industries follow a boom / bust cycle

- **Explosive growth, followed by collapse and then retrenchment**
- **Those that lived through the 'Dot COM Bubble' know this well**
- **The big mistake was to confuse Potential with Demand**
 - Anything can be sold online but not everything should be

Dot-com bubble

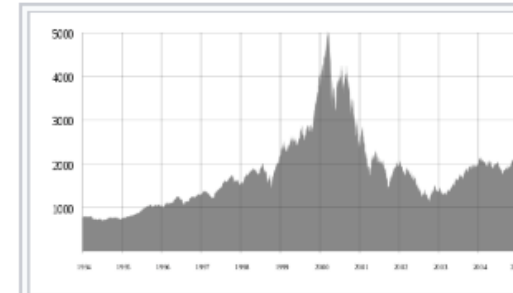
From Wikipedia, the free encyclopedia

The **dot-com bubble** (also known as the **dot-com boom**,^[1] the **tech bubble**,^[2] and the **Internet bubble**) was a historic period of excessive speculation mainly in the United States that occurred roughly from 1994 to 2000, a period of massive growth in the use and adoption of the Internet.^[3]

The [Nasdaq Composite stock market index](#), which included many Internet-based companies, peaked in value on March 10, 2000, before crashing. The burst of the bubble, known as the **dot-com crash**, lasted from March 11, 2000, to October 9, 2002.^{[3][4]} During the crash, many [online shopping](#) companies, such as [Pets.com](#), [Webvan](#), and [Boo.com](#), as well as communication companies, such as [Worldcom](#), [NorthPoint Communications](#) and [Global Crossing](#), failed and shut down.^{[5][6]} Others, such as [Cisco](#), whose stock declined by 86%,^[6] and [Qualcomm](#), lost a large portion of their market capitalization but survived, and some companies, such as [eBay](#) and [Amazon.com](#), lost value but recovered quickly.

Contents [\[hide\]](#)

1 Prelude to the bubble



The [NASDAQ Composite](#) index spiked in the late 1990s and then fell sharply as a result of the dot-com bubble.



Quarterly U.S. venture capital investments, 1995–2017



“We accept the reality of the world with which we are presented.”

– Cristof (The Truman Show)



BUSINESS
INSIDER



Join

How WeWork spiraled from a \$47 billion valuation to talk of bankruptcy in just 6 weeks

Dakin Campbell Sep 28, 2019, 2:29 PM



prepare for rain

be prepared for the next downturn

- **We have experienced 15 years of continuous growth**
 - Many in the industry have only ever known success
- **It's easy to become overconfident and complacent**
- **Build up your skillset and resume**
 - Look beyond your current domain
 - Have a network of contacts



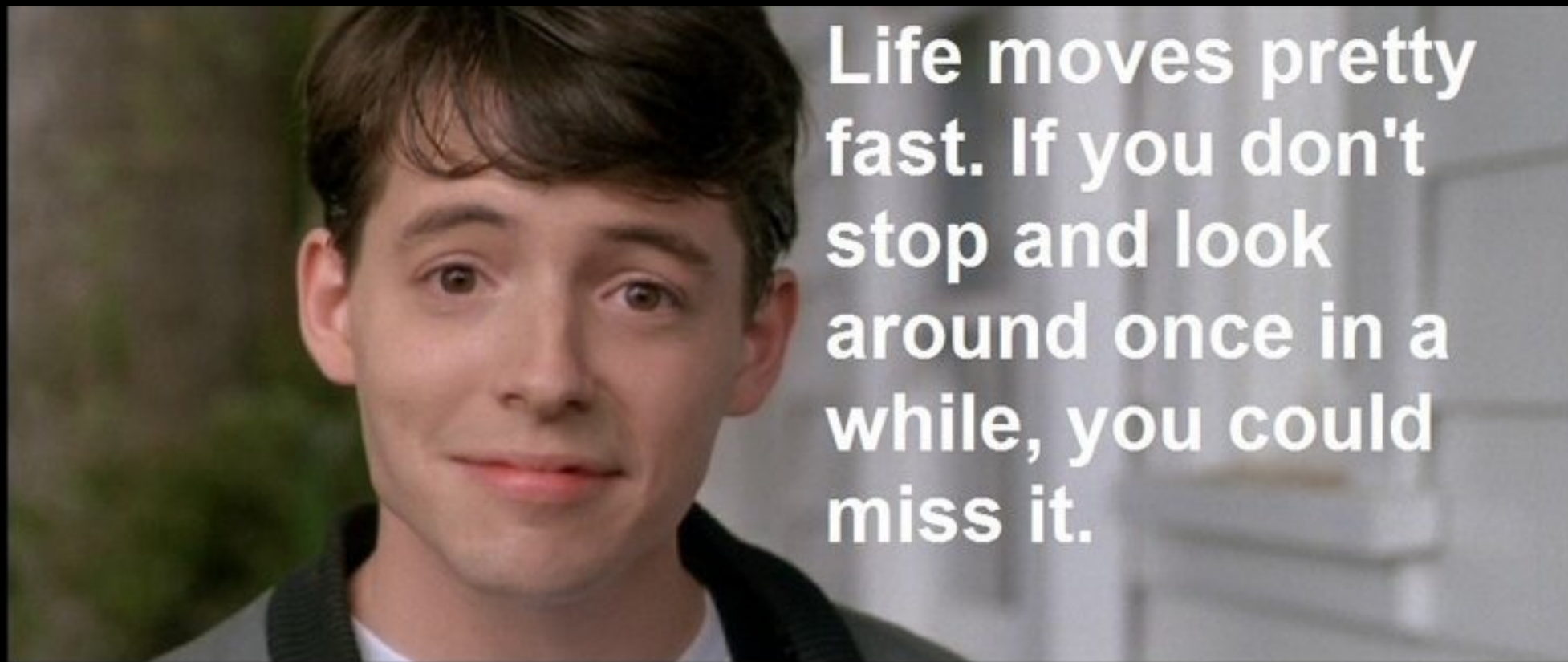
prepare for rain

what can we learn

- **prepare for extreme changes (downturn) in the industry**
- **think about your career**
- **educate yourself**
- **keep in demand**

Remember What Matters

ten



Life moves pretty fast. If you don't stop and look around once in a while, you could miss it.

remember what matters

life is short, blink and it's gone

- **this is something everyone from the 1990s knows**
- **time passes far more quickly than you think**
- **don't forget to smell the roses...**
- **have other things important to you in your life**

The 10 Lessons

Summary

our lessons from the 1990s

- 1. draw more pictures**
- 2. review at the right level**
- 3. value stability**
- 4. invent it here**
- 5. learn to test**
- 6. master the tools**
- 7. focus on fundamentals**
- 8. remain accountable**
- 9. prepare for rain**
- 10. remember what matters**





GOTO
Guide



Remember to
rate this session

THANK YOU!



#GOTOams