Software Engineering Theory vs. Practice: A Song of Ice and Tire Fire

Thought Leader, Disruptive Innovator

Senior SRE Leader at Google

Senior Software Engineer at Netflix

SVP of Thoughts at Facebook

Obviously better than you

Disclaimer: absolutely none of the above is true

Official Hiptech Translator

Native proficiency in Russian, Hebrew, English, Java, Go

Curses in 18 more languages

Fluent in Thought Leader gibberish

Disclaimer: some of the above may or may not be true.

Baruch,
Thought
Leader Away!

Everybody's software must be releasable at absolutely any time

Everyone must have 100% test automation

We do Continuous Security well.

Your greatest threat is an outage.

Not an employee.

VMs are the enemy of DevOps. This is where you must focus your innovation.

You are a beautiful unique snowflake, as are your problems.

No vendor could possibly understand them.

Our company is based in SF because that’s where the best engineers are.

RUMBABEAT
Thank you!

BARUCH SADOGURSKY

CHIEF STICKER OFFICER

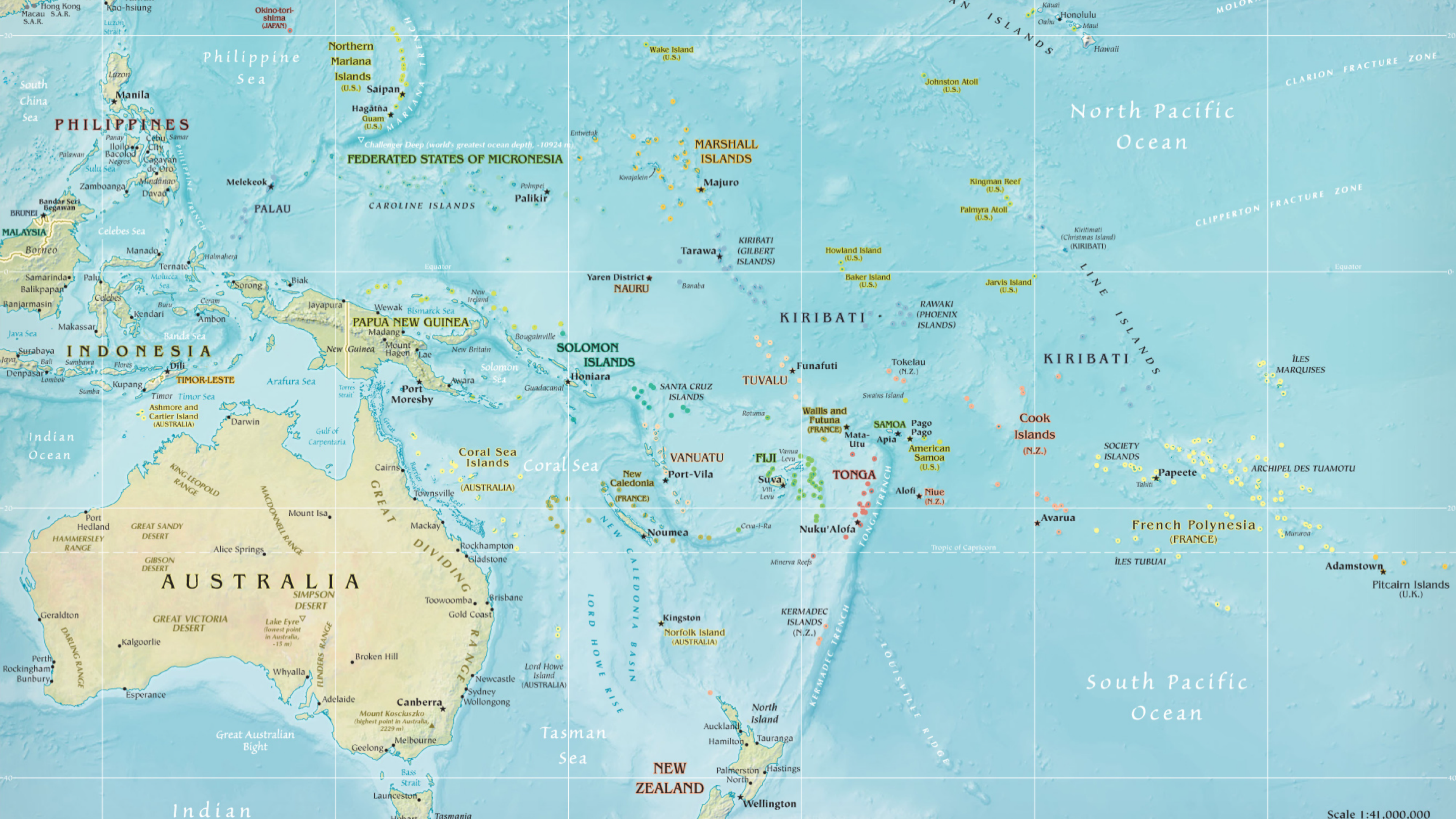
(ALSO, HEAD OF DEVELOPER RELATIONS)

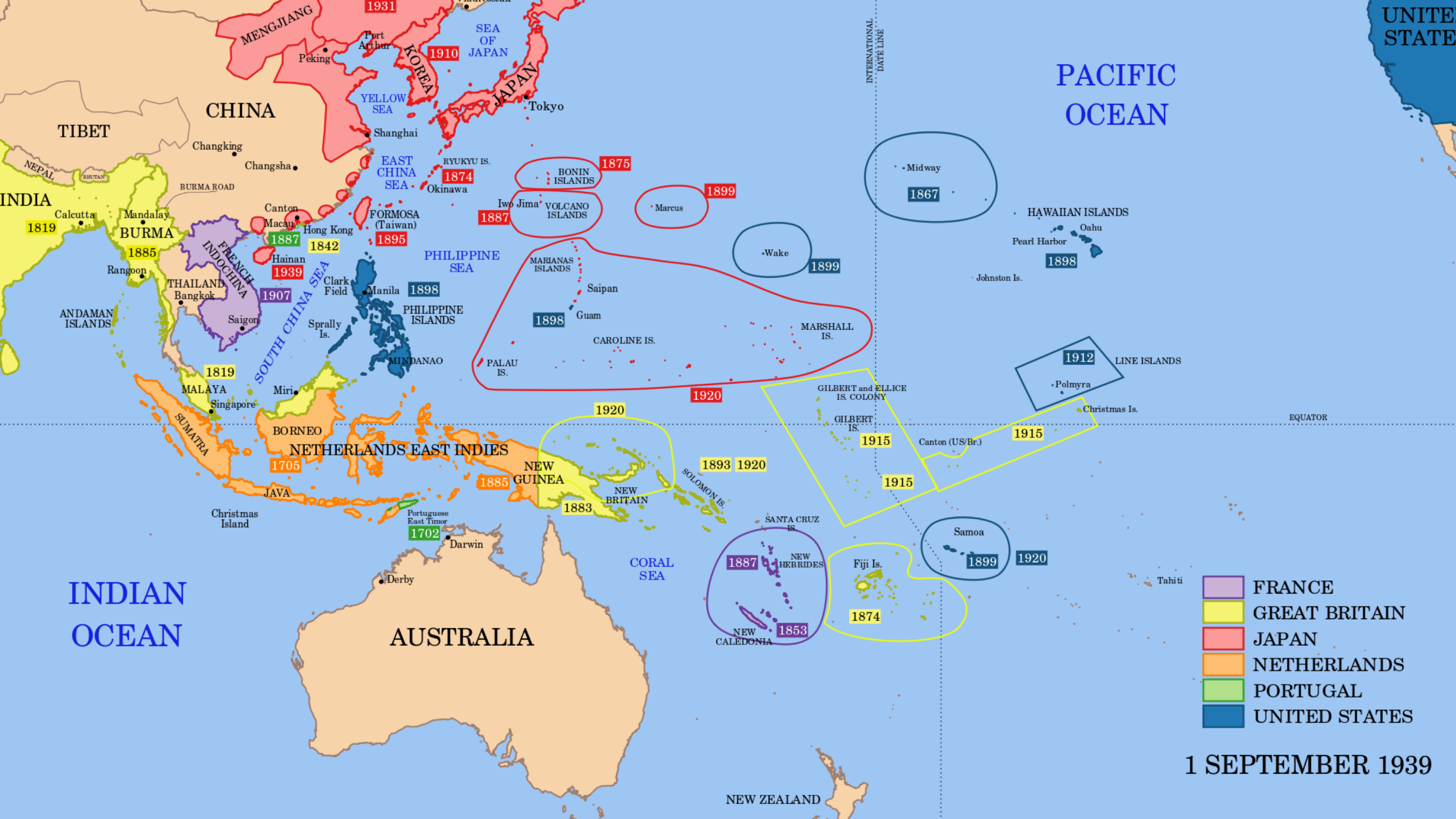
✉ JBARUCH@JFROG.COM

@JBARUCH

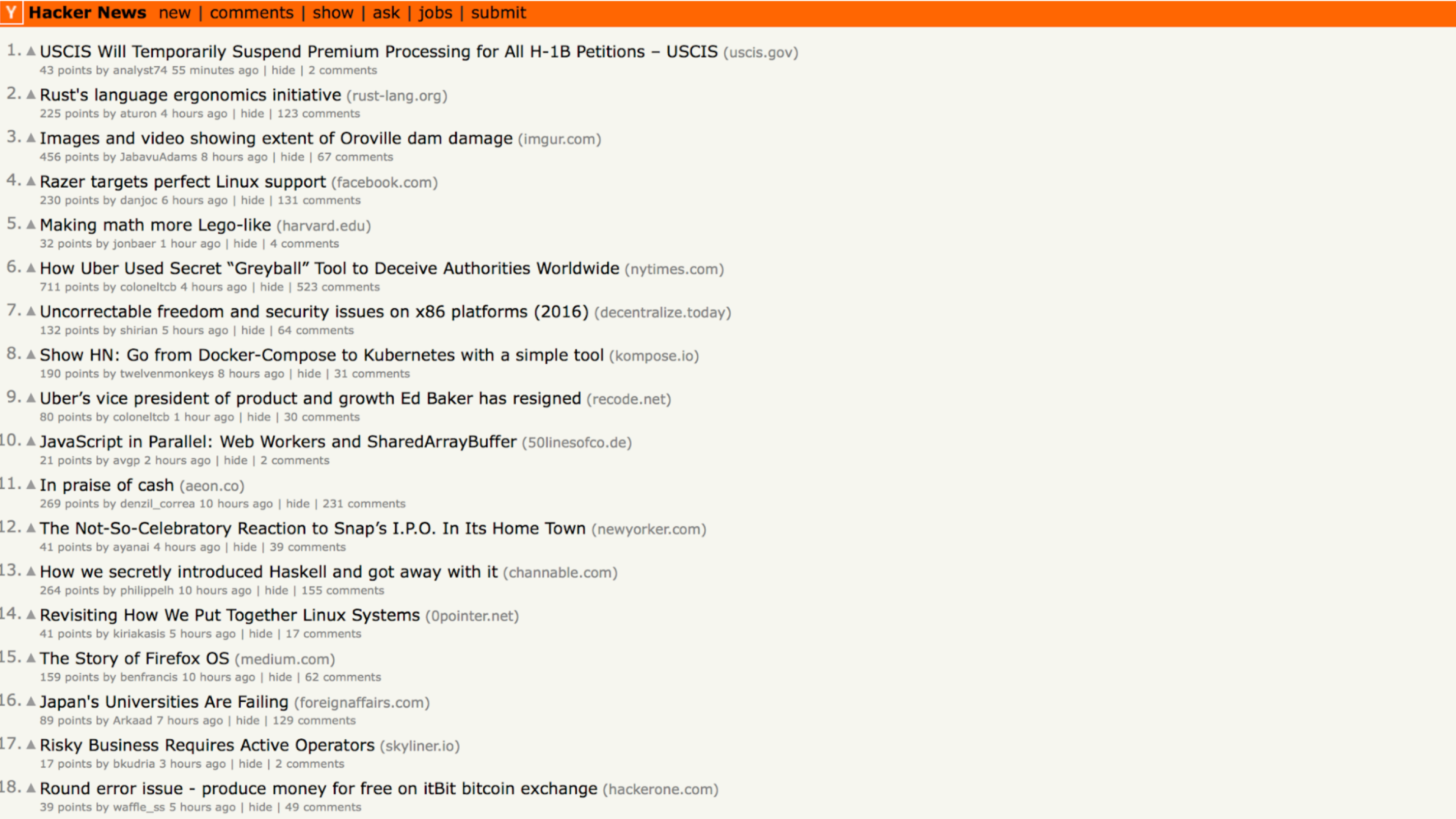
+1(408)890-9281

How did we get here?


Hong Kong S.A.R.
Macau S.A.R.
Kao-hsiung
Luzon Strait
Okino-tori-shima (JAPAN)
Northern Mariana Islands (U.S.)
Saipan
Hagåtña
Guam (U.S.)
MARIANA TRENCH
Challenger Deep (world's greatest ocean depth, -10924 m)
Philippine Sea
South China Sea
PHILIPPINES
Luzon
Manila
Panay
Cebu
Samar
Iloilo
Bacolod
Negros
City
Cagayan de Oro
Palawan
Sulu Sea
Mindanao
Zamboanga
Davao
PHILIPPINE TRENCH
Melekeok
PALAU
FEDERATED STATES OF MICRONESIA
CAROLINE ISLANDS
Pohnpei
Palikir
Wake Island (U.S.)
Enewetak
Kwajalein
MARSHALL ISLANDS
Majuro
Johnston Atoll (U.S.)
Kauai
Honolulu
Oahu
Maui
Hawaii
ISLANDS
MOLOKAI
CLARION FRACTURE ZONE
North Pacific Ocean
Kingman Reef (U.S.)
Palmyra Atoll (U.S.)
CLIPPERTON FRACTURE ZONE
Kiritimati (Christmas Island) (KIRIBATI)
LINE ISLANDS
Equator
Bandar Seri Begawan
BRUNEI
MALAYSIA
Borneo
Celebes Sea
Manado
Halmahera
Ternate
Samarinda
Palu
Molucca Sea
Balikpapan
Celebes
Sorong
Biak
Banjarmasin
Buru
Ceram
Kendari
Ambon
Java Sea
Makassar
Banda Sea
INDONESIA
Surabaya
Java
Bali
Sumbawa
Flores
Dili
Denpasar
Lombok
Sumba
Kupang
TIMOR-LESTE
Timor
Timor Sea
Ashmore and Cartier Island (AUSTRALIA)
Tarawa
KIRIBATI (GILBERT ISLANDS)
Yaren District
NAURU
Banaba
Howland Island (U.S.)
Baker Island (U.S.)
Jarvis Island (U.S.)
KIRIBATI
RAWAKI (PHOENIX ISLANDS)
Jayapura
Wewak
New Ireland
Bismarck Sea
PAPUA NEW GUINEA
Madang
New Guinea
Mount Hagen
Lae
New Britain
Bougainville
SOLOMON ISLANDS
Honiara
Solomon Sea
Guadalcanal
Arawa
Port Moresby
Torres Strait
Arafura Sea
Funafuti
TUVALU
Tokelau (N.Z.)
Swains Island
ÎLES MARQUISES
SANTA CRUZ ISLANDS
Rotuma
Wallis and Futuna (FRANCE)
Mata-Utu
SAMOA
Apia
Pago Pago
American Samoa (U.S.)
Cook Islands (N.Z.)
SOCIETY ISLANDS
Papeete
Tahiti
ARCHIPEL DES TUAMOTU
Indian Ocean
Darwin
Gulf of Carpentaria
Great Barrier Reef
Coral Sea Islands (AUSTRALIA)
Coral Sea
VANUATU
Port-Vila
FIJI
Vanua Levu
Suva
Viti Levu
TONGA
TONGA TRENCH
Alofi
Niue (N.Z.)
Cairns
KING LEOPOLD RANGE
MACDONNELL RANGE
Townsville
Mount Isa
Mackay
Port Hedland
GREAT SANDY DESERT
HAMMERSLEY RANGE
New Caledonia (FRANCE)
Noumea
NEW CALEDONIA BASIN
Ceva-i-Ra
Nuku'Alofa
Avarua
French Polynesia (FRANCE)
Mururoa
Tropic of Capricorn
ÎLES TUBUAI
Adamstown
Pitcairn Islands (U.K.)
Minerva Reefs
Alice Springs
GIBSON DESERT
Rockhampton
Gladstone
GREAT DIVIDING RANGE
AUSTRALIA
SIMPSON DESERT
Toowoomba
Brisbane
Gold Coast
Geraldton
GREAT VICTORIA DESERT
Lake Eyre (lowest point in Australia, -15 m)
Kalgoorlie
DARLING RANGE
FLINDERS RANGE
Broken Hill
Kingston
Norfolk Island (AUSTRALIA)
KERMADEC ISLANDS (N.Z.)
KERMADEC TRENCH
LOUISVILLE RIDGE
LORD HOWE RISE
Perth
Rockingham
Bunbury
Whyalla
Newcastle
Lord Howe Island (AUSTRALIA)
Esperance
Adelaide
Sydney
Wollongong
Canberra
Mount Kosciuszko (highest point in Australia, 2229 m)
Great Australian Bight
Tasman Sea
South Pacific Ocean
North Island
Auckland
Hamilton
Tauranga
Melbourne
Geelong
Bass Strait
Palmerston North
Hastings
NEW ZEALAND
Wellington
Launceston
Hobart
Tasmania
Indian
Scale 1:41,000,000


1931
MENGJIANG
Port Arthur
Peking
KOREA
1910
SEA OF JAPAN
JAPAN
Tokyo
CHINA
TIBET
YELLOW SEA
Shanghai
Changking
Changsha
NEPAL
BHUTAN
BURMA ROAD
INDIA
Calcutta
1819
Mandalay
BURMA
1885
Rangoon
EAST CHINA SEA
RYUKYU IS.
1874
Okinawa
Canton
Macau
1887
Hong Kong
1842
FORMOSA (Taiwan)
1895
Hainan
1939
FRENCH INDOCHINA
1907
THAILAND
Bangkok
Saigon
ANDAMAN ISLANDS
SOUTH CHINA SEA
Clark Field
Manila
1898
PHILIPPINE SEA
PHILIPPINE ISLANDS
Spratly Is.
MINDANAO
BONIN ISLANDS
1875
Iwo Jima
VOLCANO ISLANDS
1887
Marcus
1899
MARIANAS ISLANDS
Saipan
Guam
1898
CAROLINE IS.
PALAU IS.
MARSHALL IS.
1920
INTERNATIONAL DATE LINE
Midway
1867
Wake
1899
PACIFIC OCEAN
UNITED STATES
HAWAIIAN ISLANDS
Oahu
Pearl Harbor
1898
Johnston Is.
1912
LINE ISLANDS
Polmyra
Christmas Is.
EQUATOR
1819
MALAYA
Singapore
Miri
BORNEO
SUMATRA
NETHERLANDS EAST INDIES
1705
JAVA
Christmas Island
Portuguese East Timor
1702
Darwin
Derby
1920
NEW GUINEA
1885
NEW BRITAIN
1883
SOLOMON IS.
1893
1920
GILBERT and ELLICE IS. COLONY
GILBERT IS.
1915
1915
Canton (US/Br.)
1915
SANTA CRUZ IS.
NEW HEBRIDES
1887
NEW CALEDONIA
1853
Fiji Is.
1874
Samoa
1899
1920
Tahiti
CORAL SEA
INDIAN OCEAN
AUSTRALIA
NEW ZEALAND
FRANCE
GREAT BRITAIN
JAPAN
NETHERLANDS
PORTUGAL
UNITED STATES
1 SEPTEMBER 1939

NC957Y



Hacker News new | comments | show | ask | jobs | submit

1. ▲ USCIS Will Temporarily Suspend Premium Processing for All H-1B Petitions – USCIS (uscis.gov)
43 points by analyst74 55 minutes ago | hide | 2 comments

2. ▲ Rust's language ergonomics initiative (rust-lang.org)
225 points by aturon 4 hours ago | hide | 123 comments

3. ▲ Images and video showing extent of Oroville dam damage (imgur.com)
456 points by JabavuAdams 8 hours ago | hide | 67 comments

4. ▲ Razer targets perfect Linux support (facebook.com)
230 points by danjoc 6 hours ago | hide | 131 comments

5. ▲ Making math more Lego-like (harvard.edu)
32 points by jonbaer 1 hour ago | hide | 4 comments

6. ▲ How Uber Used Secret "Greyball" Tool to Deceive Authorities Worldwide (nytimes.com)
711 points by coloneltcb 4 hours ago | hide | 523 comments

7. ▲ Uncorrectable freedom and security issues on x86 platforms (2016) (decentralize.today)
132 points by shirian 5 hours ago | hide | 64 comments

8. ▲ Show HN: Go from Docker-Compose to Kubernetes with a simple tool (kompose.io)
190 points by twelvenmonkeys 8 hours ago | hide | 31 comments

9. ▲ Uber's vice president of product and growth Ed Baker has resigned (recode.net)
80 points by coloneltcb 1 hour ago | hide | 30 comments

10. ▲ JavaScript in Parallel: Web Workers and SharedArrayBuffer (50linesofco.de)
21 points by avgp 2 hours ago | hide | 2 comments

11. ▲ In praise of cash (aeon.co)
269 points by denzil_correa 10 hours ago | hide | 231 comments

12. ▲ The Not-So-Celebratory Reaction to Snap's I.P.O. In Its Home Town (newyorker.com)
41 points by ayanai 4 hours ago | hide | 39 comments

13. ▲ How we secretly introduced Haskell and got away with it (channable.com)
264 points by philippelh 10 hours ago | hide | 155 comments

14. ▲ Revisiting How We Put Together Linux Systems (0pointer.net)
41 points by kiriakasis 5 hours ago | hide | 17 comments

15. ▲ The Story of Firefox OS (medium.com)
159 points by benfrancis 10 hours ago | hide | 62 comments

16. ▲ Japan's Universities Are Failing (foreignaffairs.com)
89 points by Arkaad 7 hours ago | hide | 129 comments

17. ▲ Risky Business Requires Active Operators (skyliner.io)
17 points by bkudria 3 hours ago | hide | 2 comments

18. ▲ Round error issue - produce money for free on itBit bitcoin exchange (hackerone.com)
39 points by waffle_ss 5 hours ago | hide | 49 comments

Cargo Cult

The Four Questions

The Four Questions

1. Is my organization/team ready to adopt a new tech?
2. Is it even a good tech?
3. What do I gain from adopting this tech?
4. Is this tech a good solution to my problem?

The Four Questions

1. Is my organization/team ready to adopt a new tech?
2. Is it even a good tech?
3. ~~What do I gain from adopting this tech?~~
4. ~~Is this tech a good solution to my problem?~~

1. Is my organization /team ready to adopt a new tech?

Introducing maturity models

"A maturity model is a tool that helps people assess the current effectiveness of a person or group and supports figuring out what capabilities they need to acquire next in order to improve their performance.

In many circles maturity models have gained a bad reputation, but although they can easily be misused, in proper hands they can be helpful."

- Martin Fowler

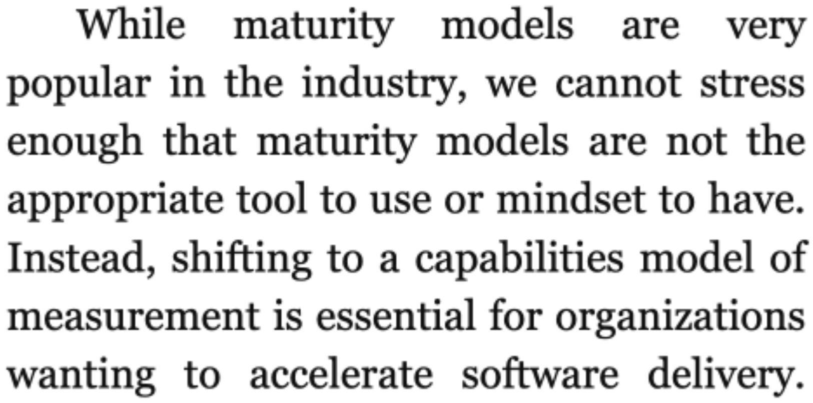
Introducing maturity models

"A maturity model is a tool that helps people assess the current effectiveness of a person or group and supports figuring out what capabilities they need to acquire next in order to improve their performance.

In many circles maturity models have gained a bad reputation, but although they can easily be misused, in proper hands they can be helpful."

- Martin Fowler

Introducing maturity models



While maturity models are very popular in the industry, we cannot stress enough that maturity models are not the appropriate tool to use or mindset to have. Instead, shifting to a capabilities model of measurement is essential for organizations wanting to accelerate software delivery.

Nicole Forsgren, Jezz Hamble, Gene Kim

Maturity Models are Bad.

Bad Maturity Models	Good Maturity Models
□ Goal	□ Process
□ Prescribed by the book	□ One size doesn't fit all
□ Checkboxes for tools	□ Focus on outcomes
□ Write and forget	□ Constantly evolve

Maturity model components

Evaluation factors

Scoring methodology

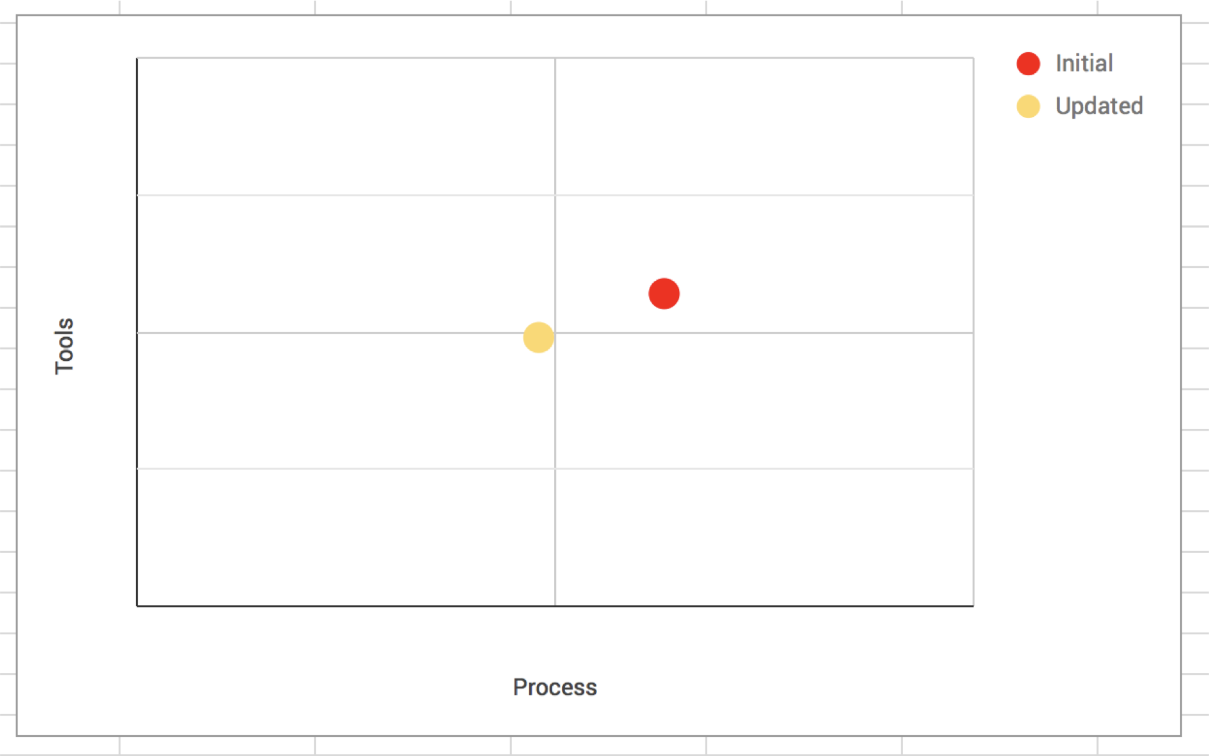
Self assessment vs 3rd party assessment capability

Progress tracking

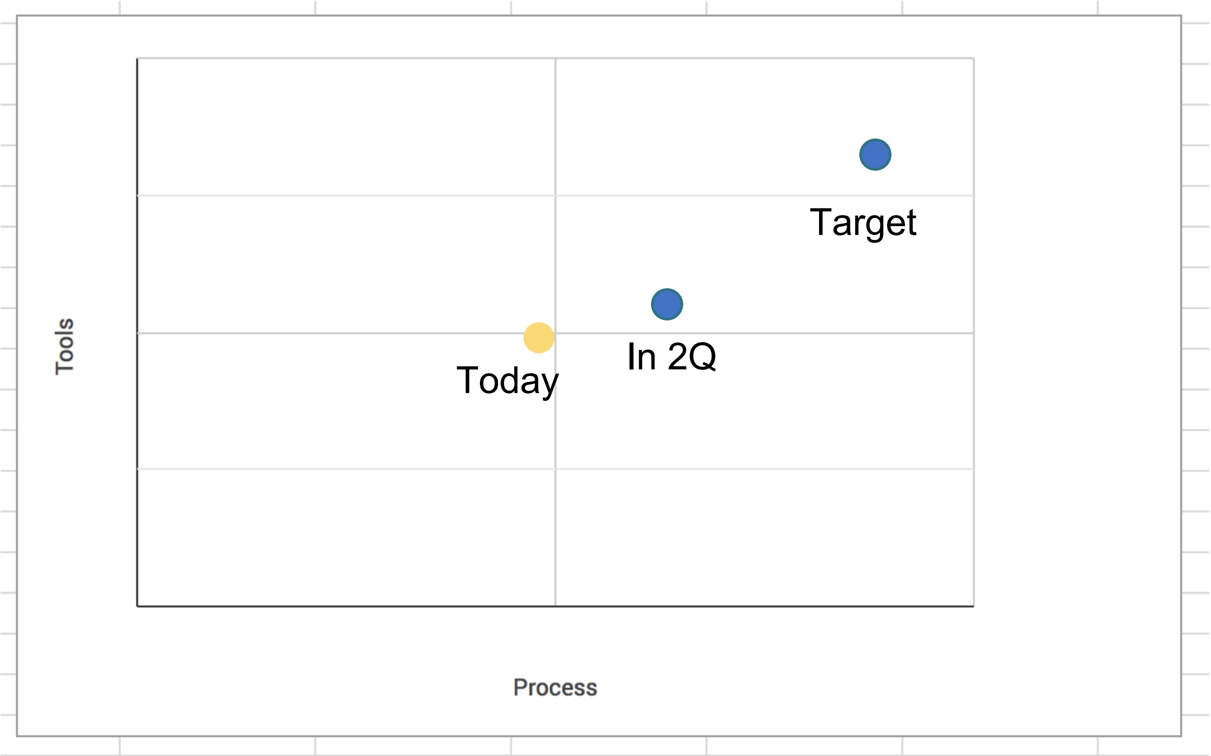
Visualization

Maturity Model Examples

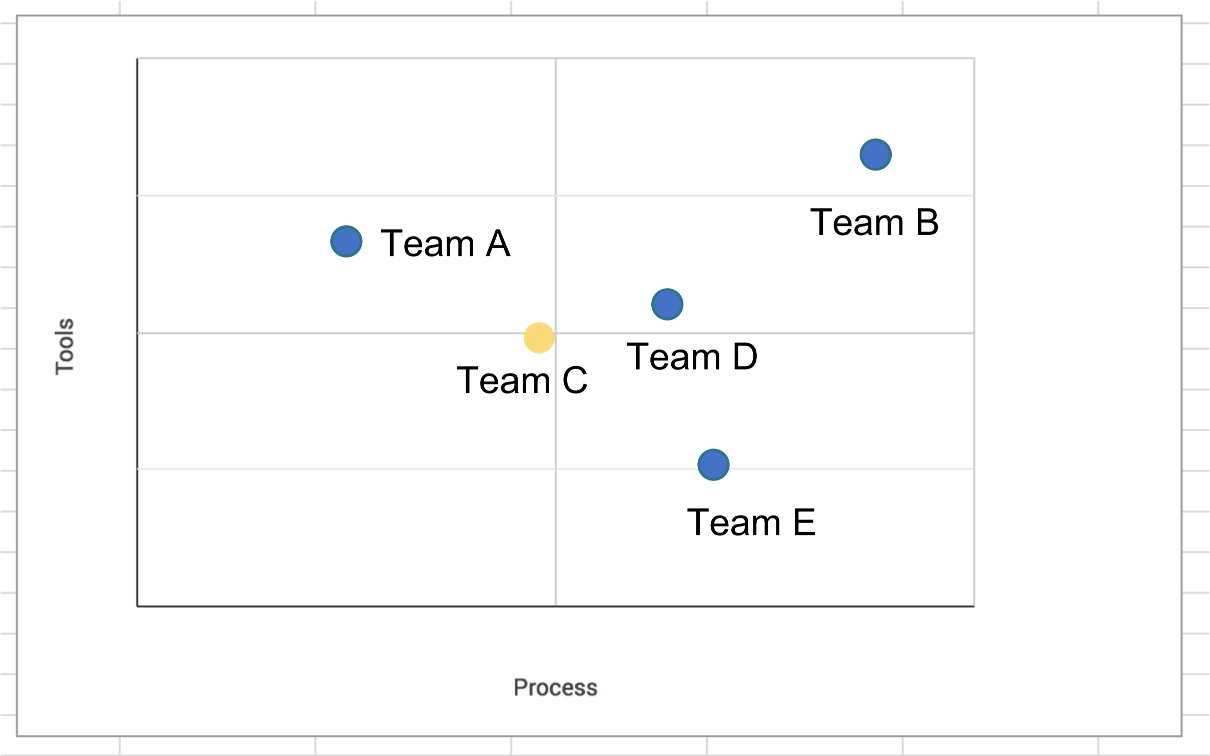
Simple model

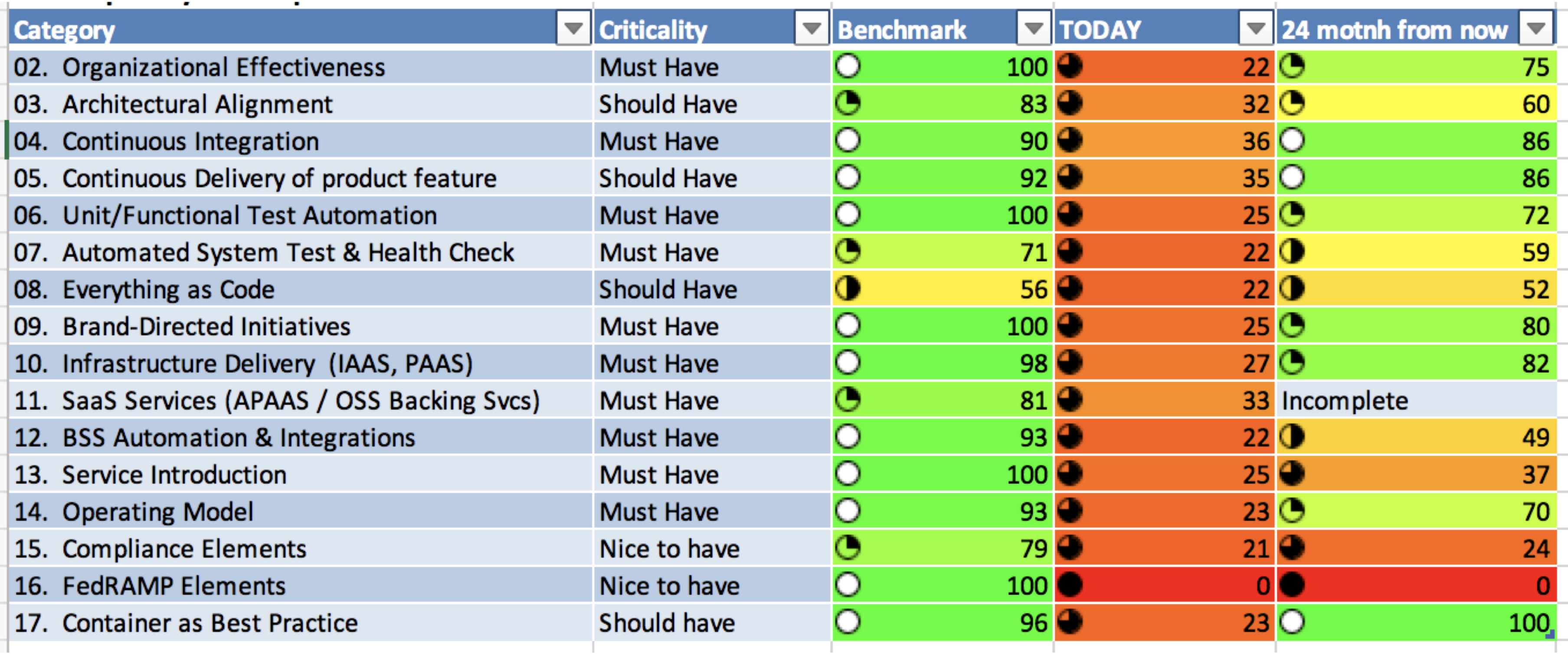


Progress planning

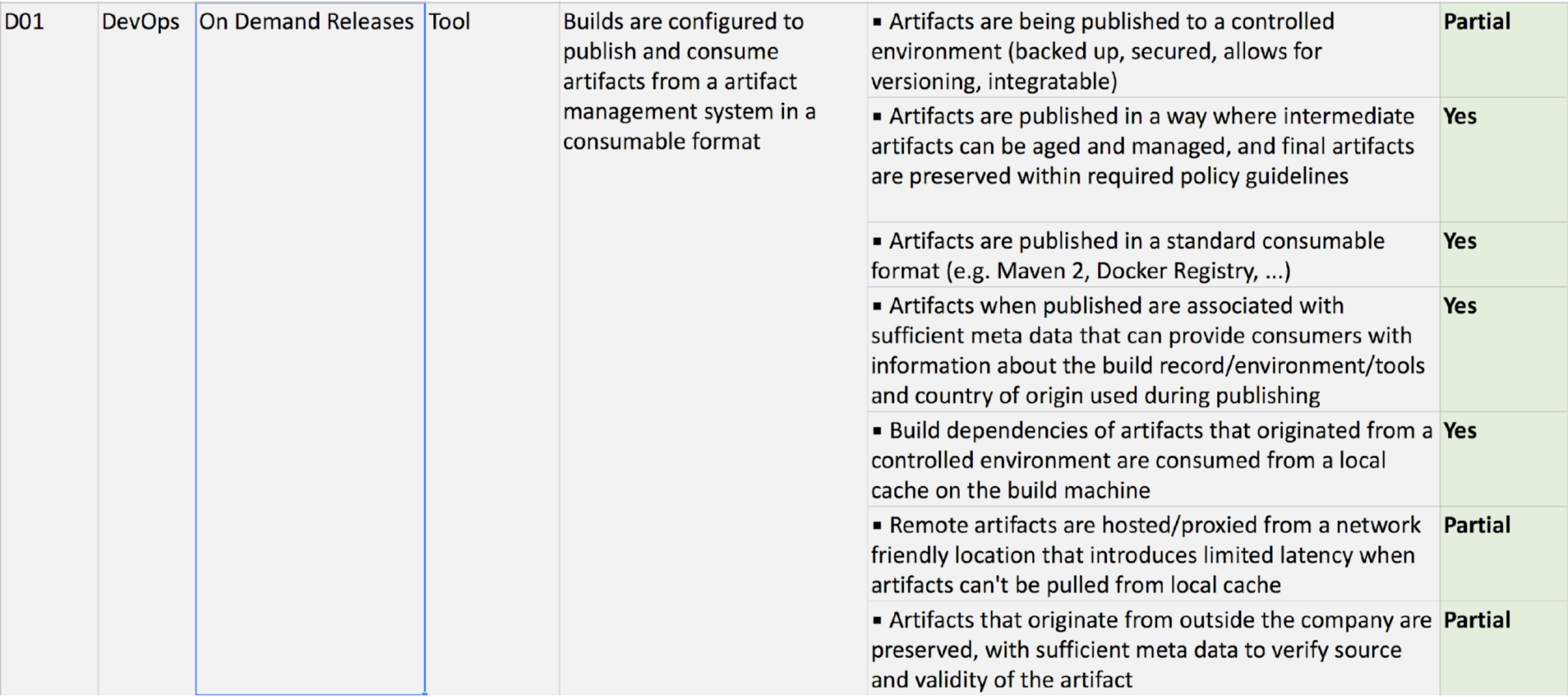


Leader board

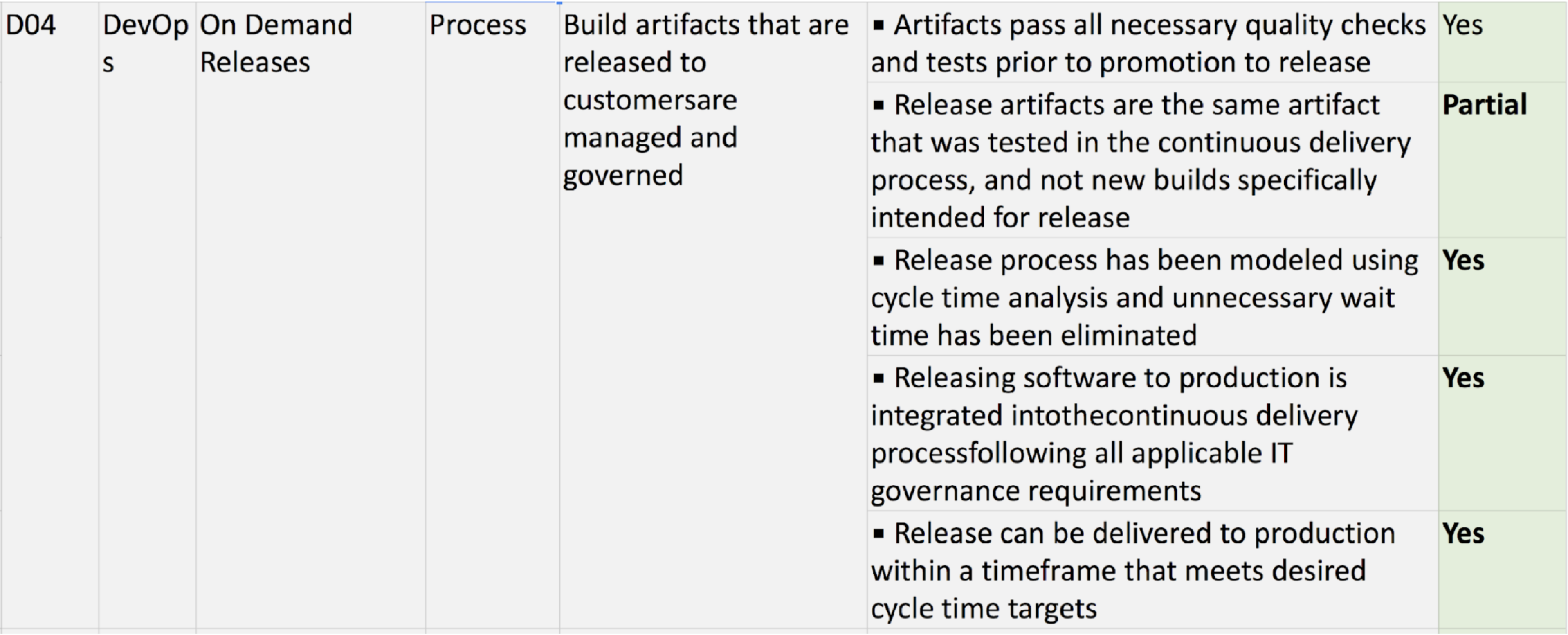




Category	Criticality	Benchmark		TODAY		24 motnh from now	
02. Organizational Effectiveness	Must Have	○	100	◕	22	◔	75
03. Architectural Alignment	Should Have	◔	83	◕	32	◔	60
04. Continuous Integration	Must Have	○	90	◕	36	○	86
05. Continuous Delivery of product feature	Should Have	○	92	◕	35	○	86
06. Unit/Functional Test Automation	Must Have	○	100	◕	25	◔	72
07. Automated System Test & Health Check	Must Have	◔	71	◕	22	◑	59
08. Everything as Code	Should Have	◑	56	◕	22	◑	52
09. Brand-Directed Initiatives	Must Have	○	100	◕	25	◔	80
10. Infrastructure Delivery (IAAS, PAAS)	Must Have	○	98	◕	27	◔	82
11. SaaS Services (APAAS / OSS Backing Svcs)	Must Have	◔	81	◕	33	Incomplete	
12. BSS Automation & Integrations	Must Have	○	93	◕	22	◑	49
13. Service Introduction	Must Have	○	100	◕	25	◕	37
14. Operating Model	Must Have	○	93	◕	23	◔	70
15. Compliance Elements	Nice to have	◔	79	◕	21	◕	24
16. FedRAMP Elements	Nice to have	○	100	●	0	●	0
17. Container as Best Practice	Should have	○	96	◕	23	○	100

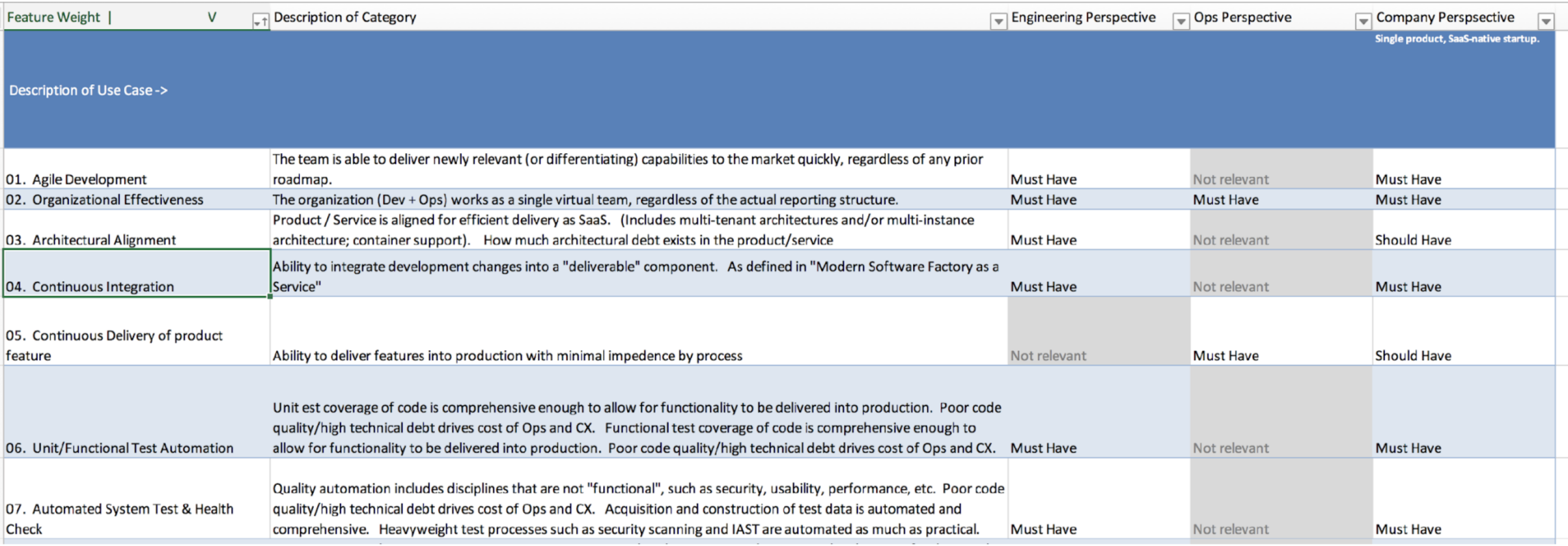


D01	DevOps	On Demand Releases	Tool	Builds are configured to publish and consume artifacts from a artifact management system in a consumable format	▪ Artifacts are being published to a controlled environment (backed up, secured, allows for versioning, integratable)	**Partial**
					▪ Artifacts are published in a way where intermediate artifacts can be aged and managed, and final artifacts are preserved within required policy guidelines	**Yes**
					▪ Artifacts are published in a standard consumable format (e.g. Maven 2, Docker Registry, ...)	**Yes**
					▪ Artifacts when published are associated with sufficient meta data that can provide consumers with information about the build record/environment/tools and country of origin used during publishing	**Yes**
					▪ Build dependencies of artifacts that originated from a controlled environment are consumed from a local cache on the build machine	**Yes**
					▪ Remote artifacts are hosted/proxied from a network friendly location that introduces limited latency when artifacts can't be pulled from local cache	**Partial**
					▪ Artifacts that originate from outside the company are preserved, with sufficient meta data to verify source and validity of the artifact	**Partial**



D04	DevOps	On Demand Releases	Process	Build artifacts that are released to customersare managed and governed	▪ Artifacts pass all necessary quality checks and tests prior to promotion to release	Yes
					▪ Release artifacts are the same artifact that was tested in the continuous delivery process, and not new builds specifically intended for release	**Partial**
					▪ Release process has been modeled using cycle time analysis and unnecessary wait time has been eliminated	**Yes**
					▪ Releasing software to production is integrated intothecontinuous delivery processfollowing all applicable IT governance requirements	**Yes**
					▪ Release can be delivered to production within a timeframe that meets desired cycle time targets	**Yes**

Account for different teams' priorities



Feature Weight | V	Description of Category	Engineering Perspective	Ops Perspective	Company Perpspective
				Single product, SaaS-native startup.
Description of Use Case ->				
01. Agile Development	The team is able to deliver newly relevant (or differentiating) capabilities to the market quickly, regardless of any prior roadmap.	Must Have	Not relevant	Must Have
02. Organizational Effectiveness	The organization (Dev + Ops) works as a single virtual team, regardless of the actual reporting structure.	Must Have	Must Have	Must Have
03. Architectural Alignment	Product / Service is aligned for efficient delivery as SaaS. (Includes multi-tenant architectures and/or multi-instance architecture; container support). How much architectural debt exists in the product/service	Must Have	Not relevant	Should Have
04. Continuous Integration	Ability to integrate development changes into a "deliverable" component. As defined in "Modern Software Factory as a Service"	Must Have	Not relevant	Must Have
05. Continuous Delivery of product feature	Ability to deliver features into production with minimal impedence by process	Not relevant	Must Have	Should Have
06. Unit/Functional Test Automation	Unit est coverage of code is comprehensive enough to allow for functionality to be delivered into production. Poor code quality/high technical debt drives cost of Ops and CX. Functional test coverage of code is comprehensive enough to allow for functionality to be delivered into production. Poor code quality/high technical debt drives cost of Ops and CX.	Must Have	Not relevant	Must Have
07. Automated System Test & Health Check	Quality automation includes disciplines that are not "functional", such as security, usability, performance, etc. Poor code quality/high technical debt drives cost of Ops and CX. Acquisition and construction of test data is automated and comprehensive. Heavyweight test processes such as security scanning and IAST are automated as much as practical.	Must Have	Not relevant	Must Have

Model definition example

System config as Code	The infrastructure configuration is managed as code - e.g. no manual processes for configuring/setting up/ infrastructure. Differentiating: Infrastructure operates without any manual processes. All changes to the infrastructure or infrastructure capabilities are done through automation and policy only. Complete: Infrastructure operates without any manual processes. Some infrequent administrative activities may be initiated manually (although the activities themselves must be automated). Partial (Most): Infrastructure operates without any manual processes. Some infrequent administrative activities may be manual, pending automation. Partial (Much): Infrastructure operates with significant automation. Some processes still manual; pending automation. Partial (Some): Infrastructure requires significant care and feeding. Many processes still manual; pending automation. No Support: While some functions may be automated, they are generally kicked-off manually; and many functions are still fully manual. Large backlog of automation items.

Applying maturity models: DOs and DONT's

Only use primary colors

Involve your teams in the model definition

Let team self assess first and then assess together

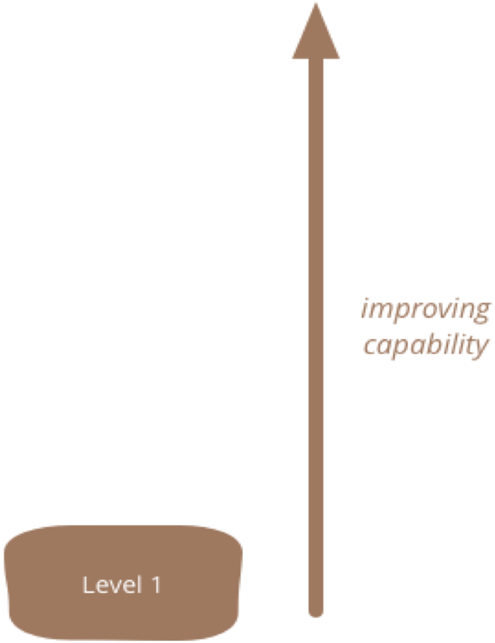
Partner with forward looking teams first

Remember being at 100% is not a goal the model has to have a stretch goal

Evolve the model from time to time

And ….

Our message is:



https://martinfowler.com/bliki/MaturityModel.html

2. *Is it even a good tech?*

FORRESTER®

IDC

Gartner

IHS Markit®

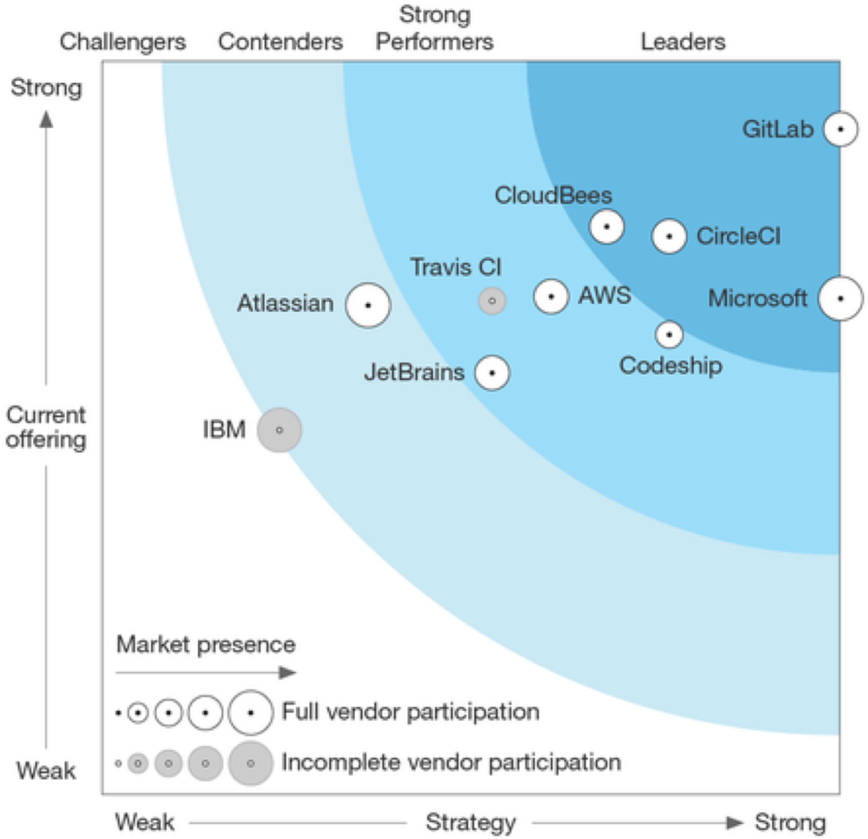
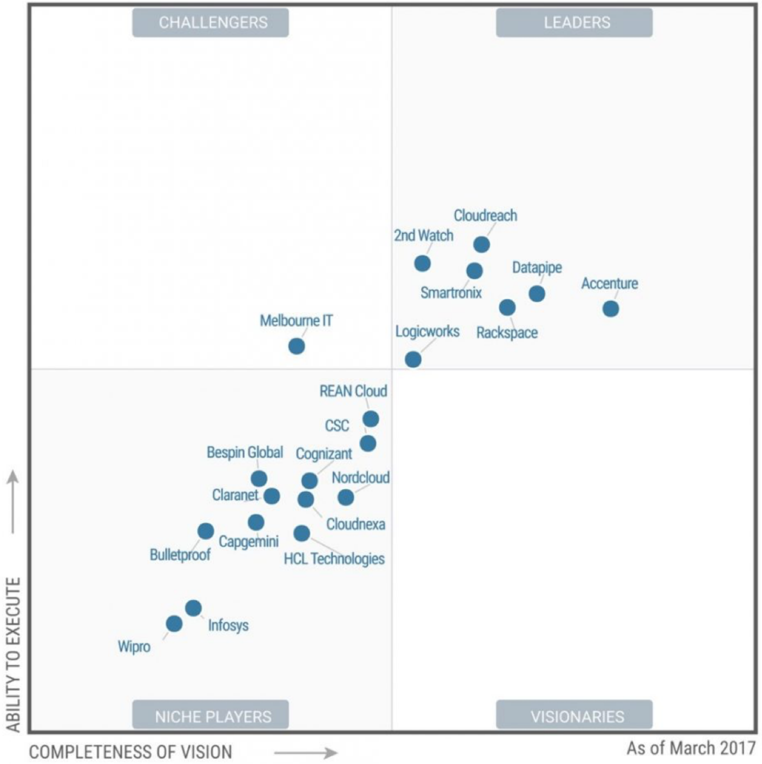
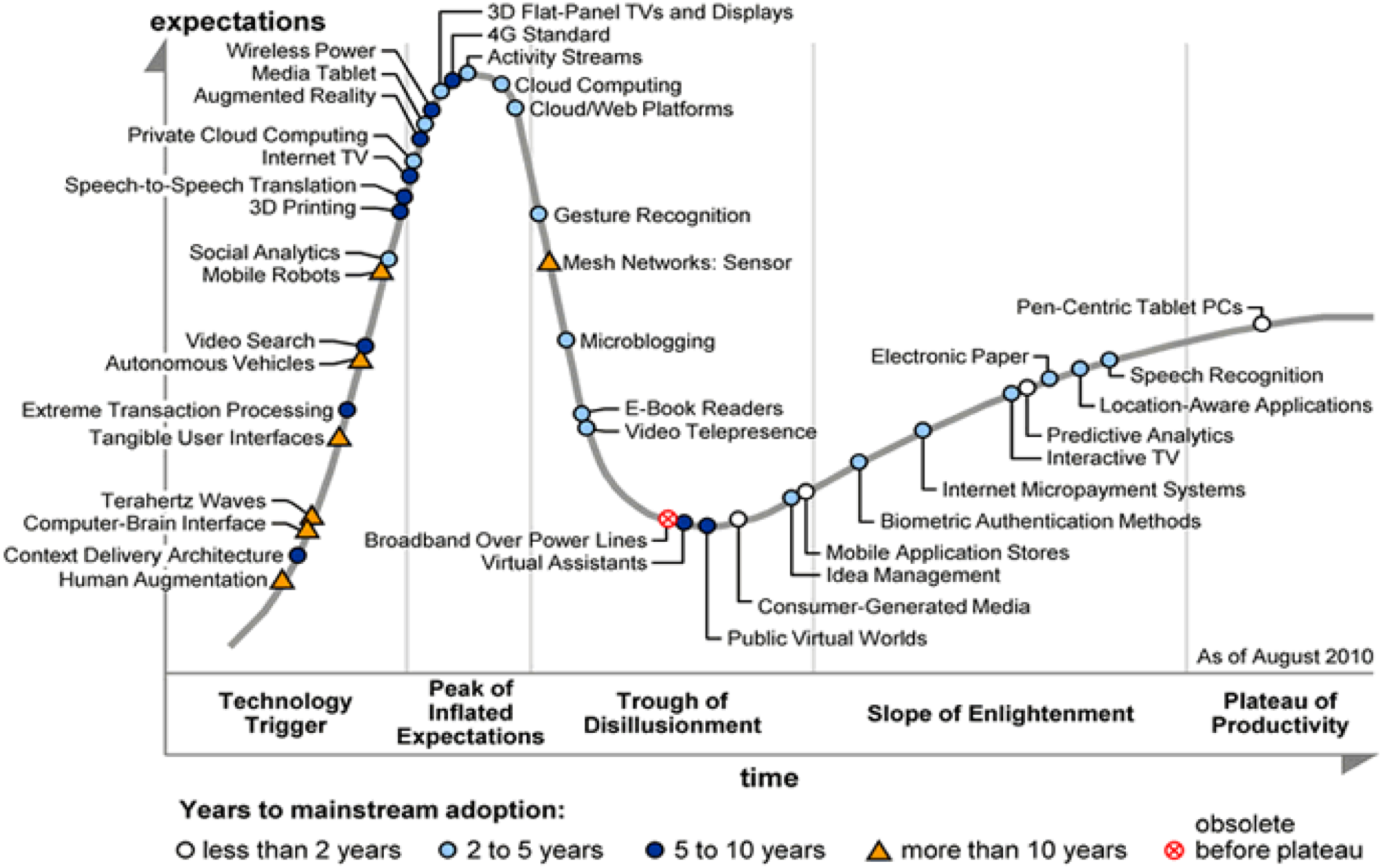


Figure 1. Magic Quadrant for Public Cloud Infrastructure Managed Service Providers, Worldwide

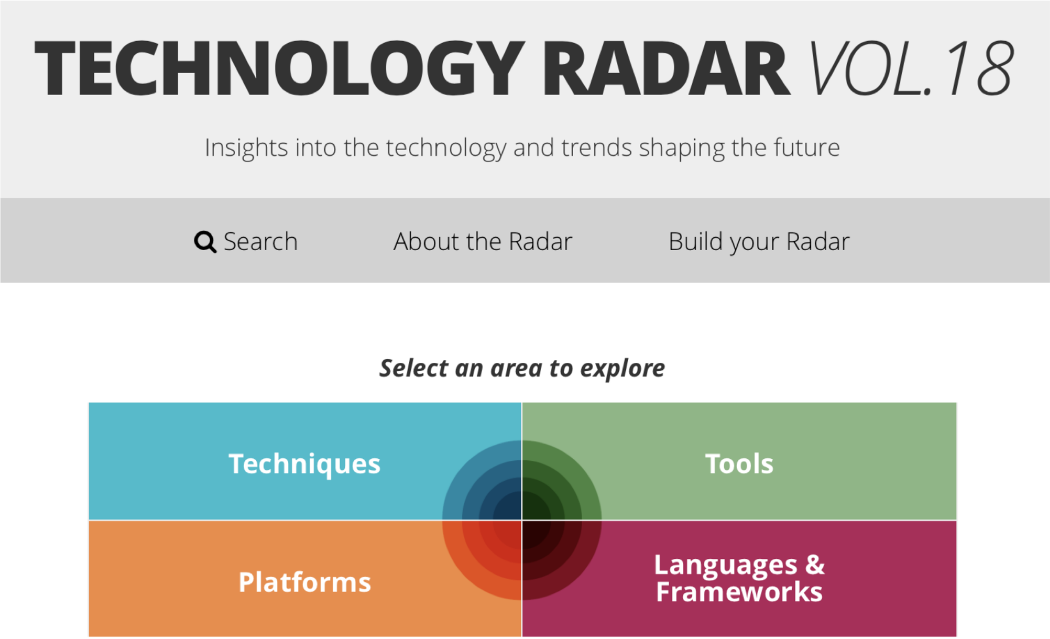


Source: Gartner (March 2017)



@jbaruch @nashvilledevops #DevOpsDays jfrog.com/shownotes

ThoughtWorks®

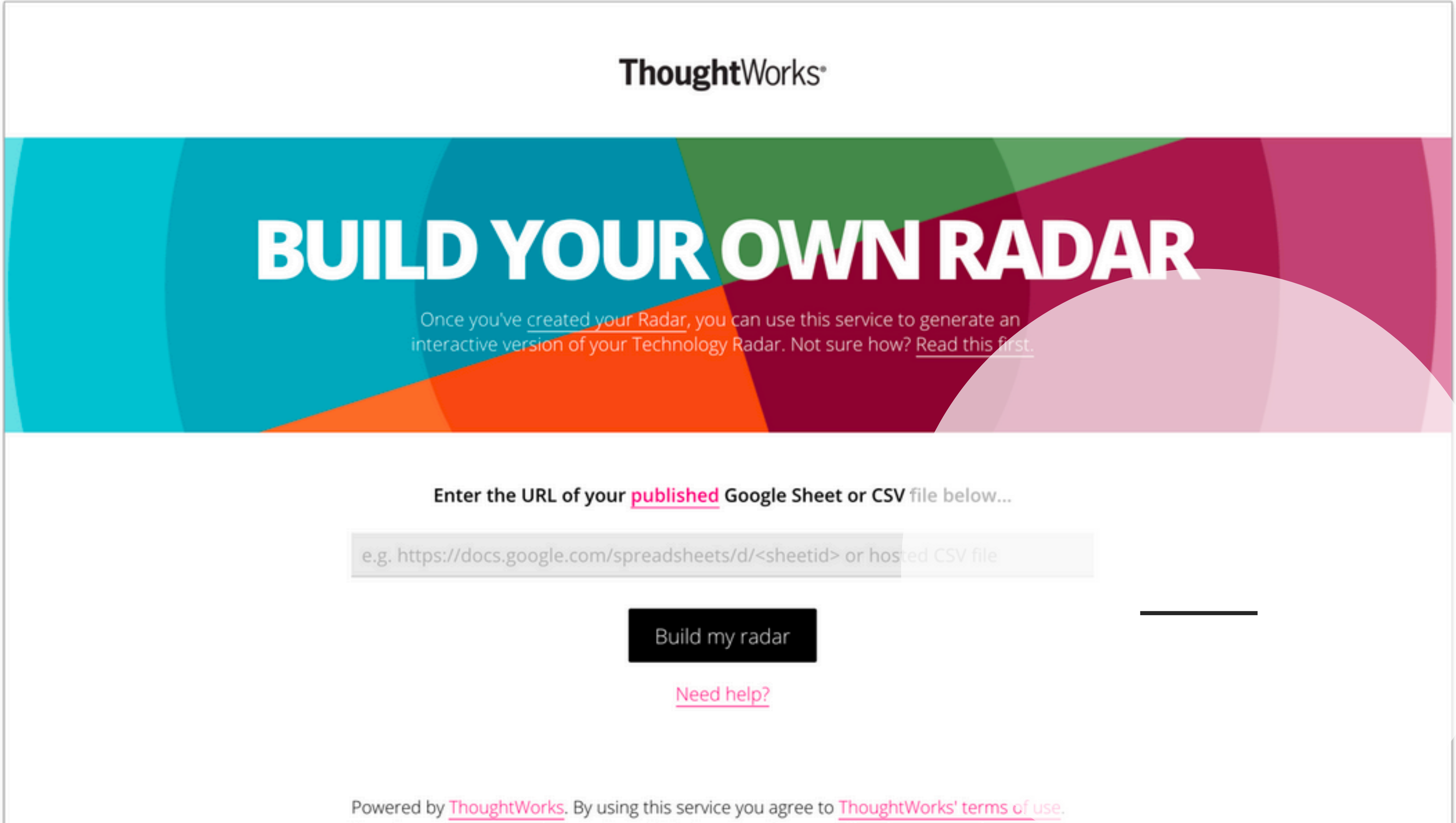


TECHNOLOGY RADAR *VOL.18*

Insights into the technology and trends shaping the future

Select an area to explore

Can I have my own?



ThoughtWorks

BUILD YOUR OWN RADAR

Once you've created your Radar, you can use this service to generate an interactive version of your Technology Radar. Not sure how? Read this first.

Enter the URL of your published Google Sheet or CSV file below...

e.g. https://docs.google.com/spreadsheets/d/<sheetid> or hosted CSV file

Powered by ThoughtWorks. By using this service you agree to ThoughtWorks' terms of use.

The Recap: The Four Questions

1. Is my organization/team ready to adopt a new tech?
2. Is it even a good tech?
3. What do I gain from adopting this tech?
4. Is this tech a good solution to my problem?