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
ALUMNI

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.

@jbaruch @austindevops jfrog.com/shownotes

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.

@jbaruch @austindevops jfrog.com/shownotes

BARUCH SADOGURSKY

CHIEF STICKER OFFICER

(ALSO, HEAD OF DEVELOPER RELATIONS)

JBARUCH@JFROG.COM

@JBARUCH

+1(408)890-9281

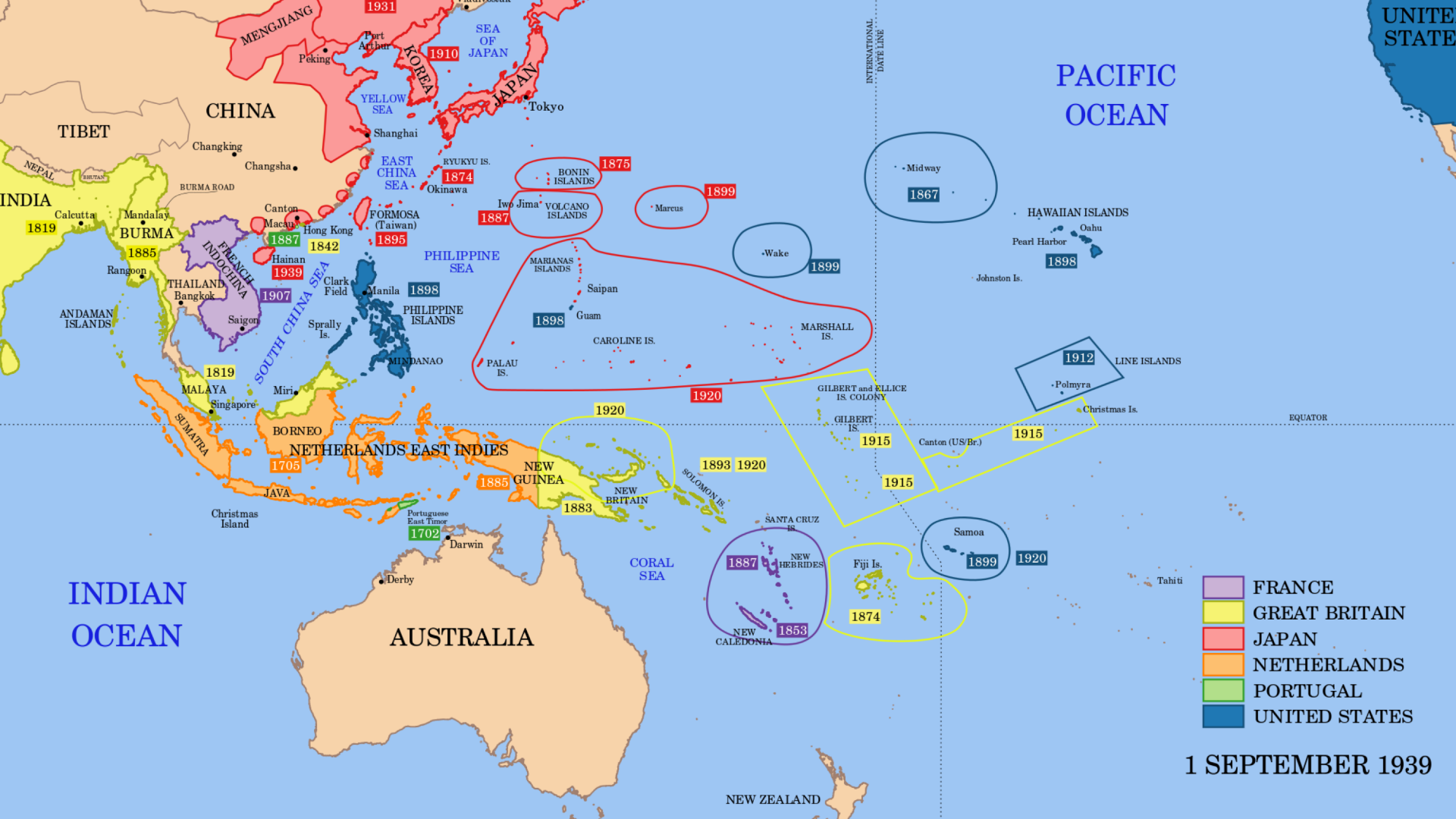
JFrog

Shownotes!

jfrog.com/shownotes
Slides
Video (by tomorrow)
All the links!
Comments, Ratings
Raffle!

How did we get here?

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



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

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

@jbaruch @austindevops jfrog.com/shownotes

The Four Questions

@jbaruch @austindevops jfrog.com/shownotes

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

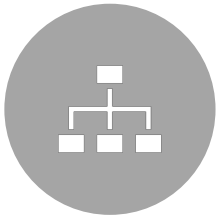
Introducing maturity models

> 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

Bad Maturity Models are Bad.

Maturity is a process, not a goal



One size does not fit all organizations (or teams in organization)

Focus on outcomes, not existence of a process or a tool

Always evolve the model as the industry and the organization evolves

Maturity model components

Evaluation factors

Scoring methodology

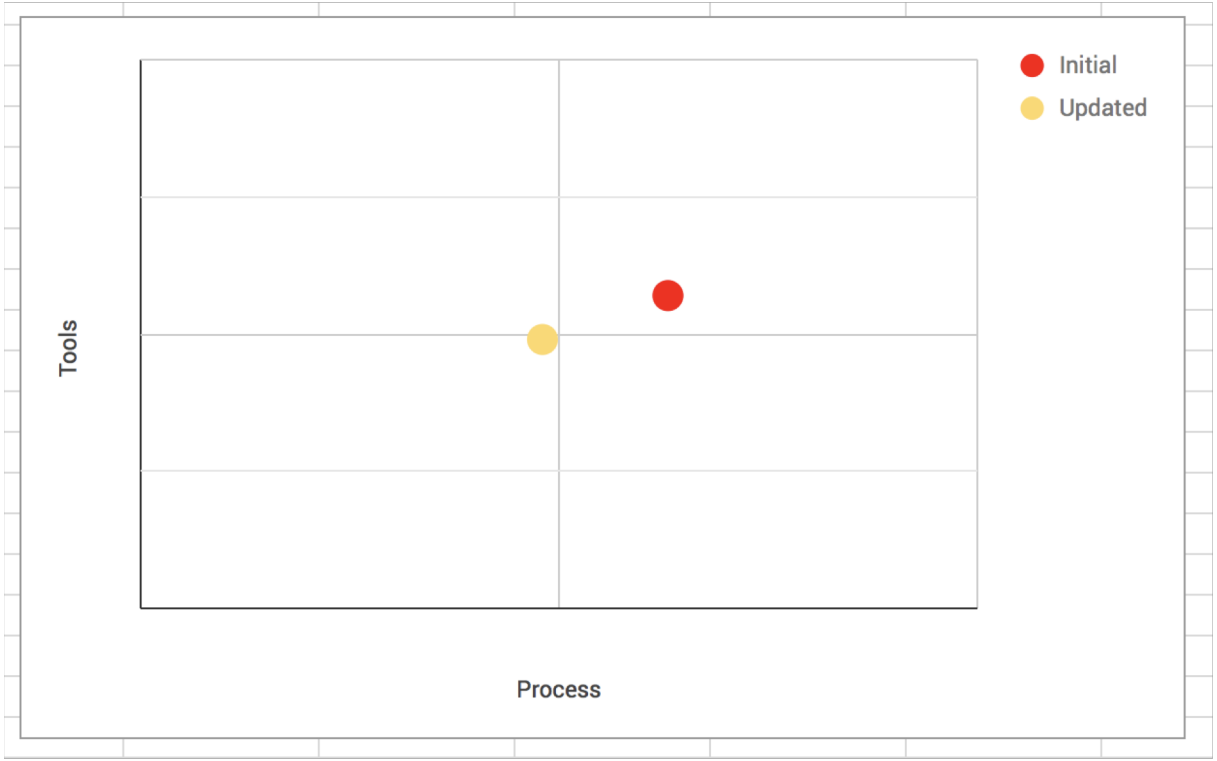
Self assessment vs 3rd party assessment capability

Progress tracking

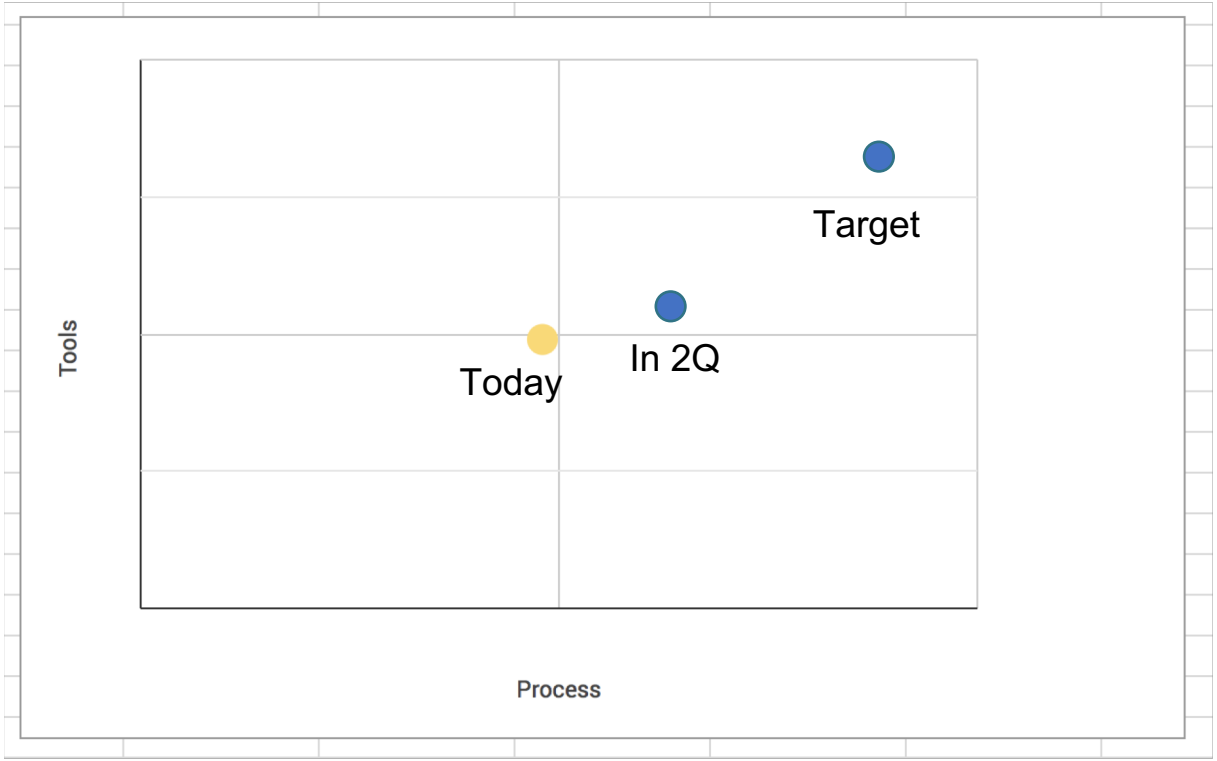
Visualization

Maturity Model Examples

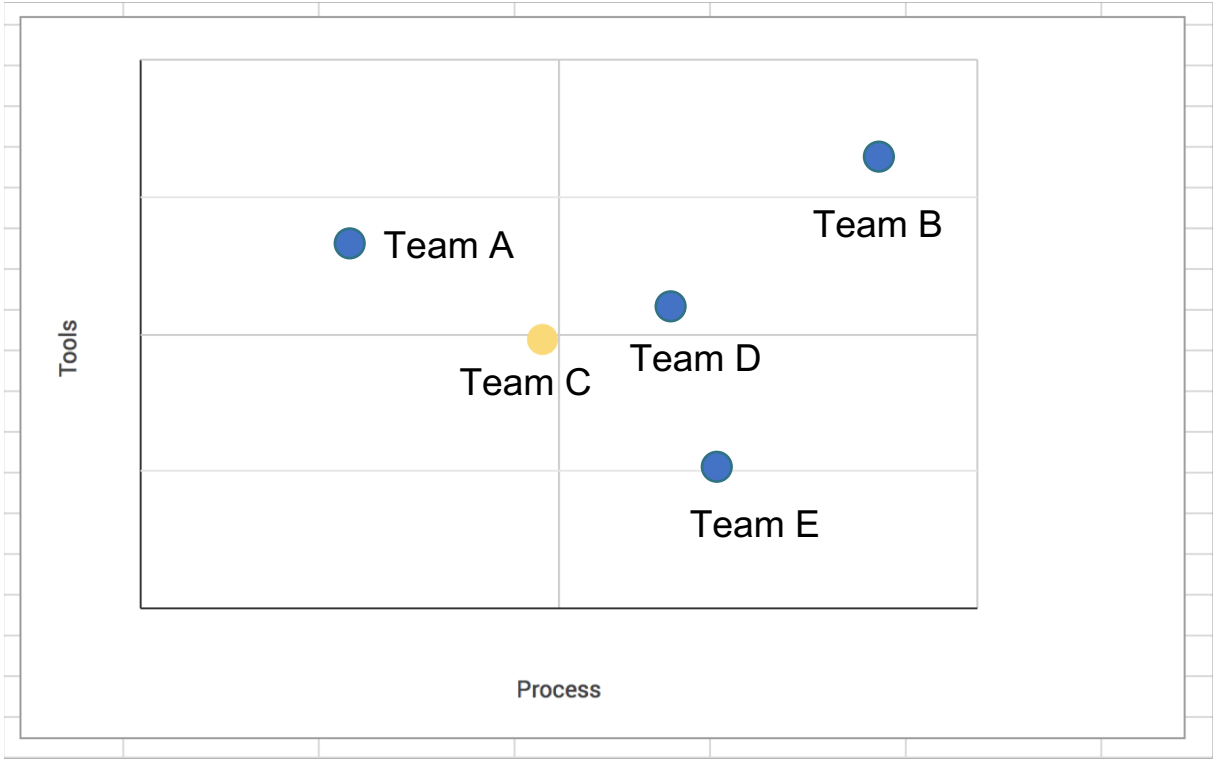
Simple model



Progress planning

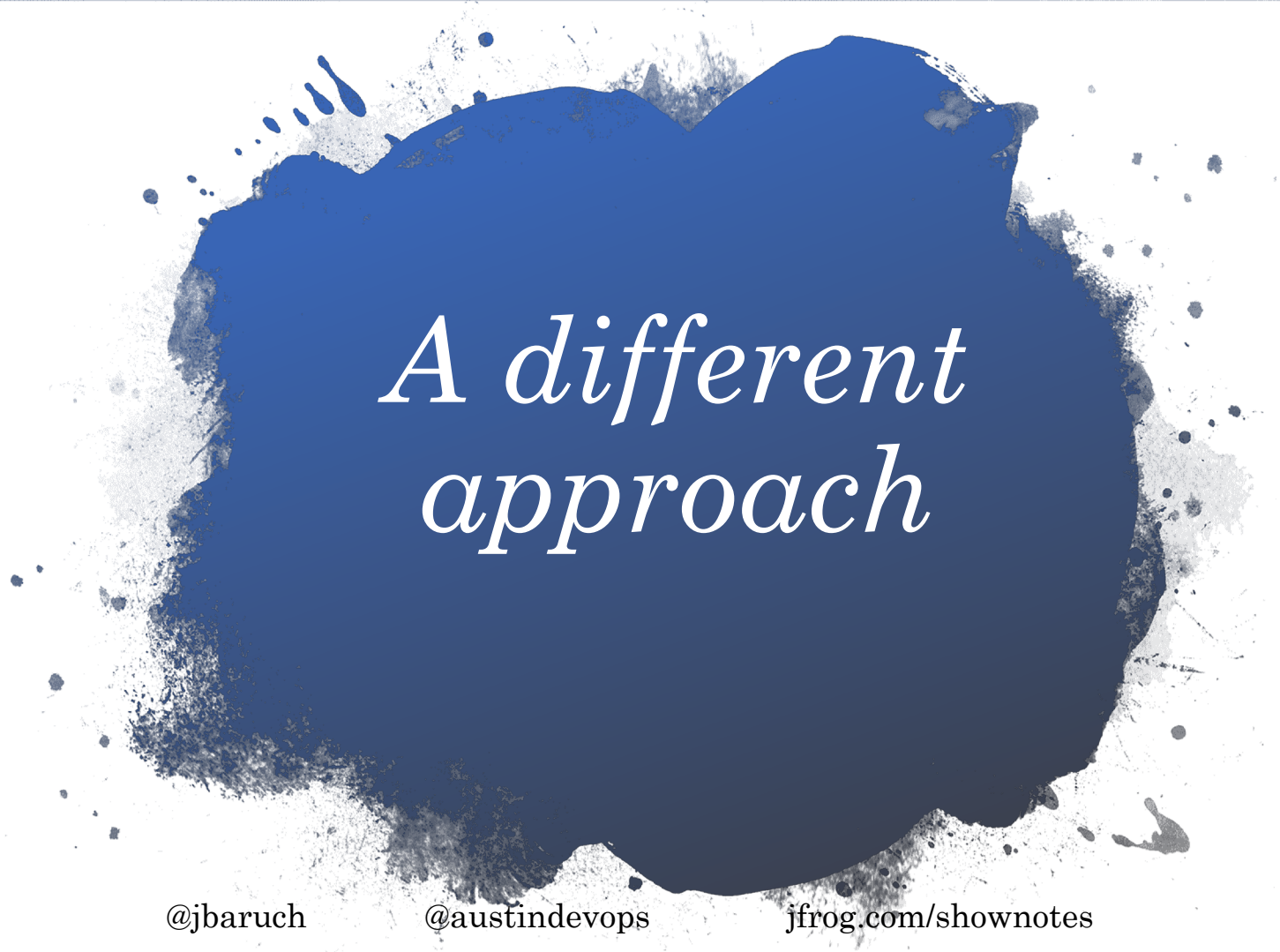


Leader board



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**


A different approach

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

Account for different teams' priorities

Feature Weight | V	Description of Category	Engineering Perspective	Ops Perspective	Company Perspsective
Description of Use Case ->				Single product, SaaS-native startup.
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

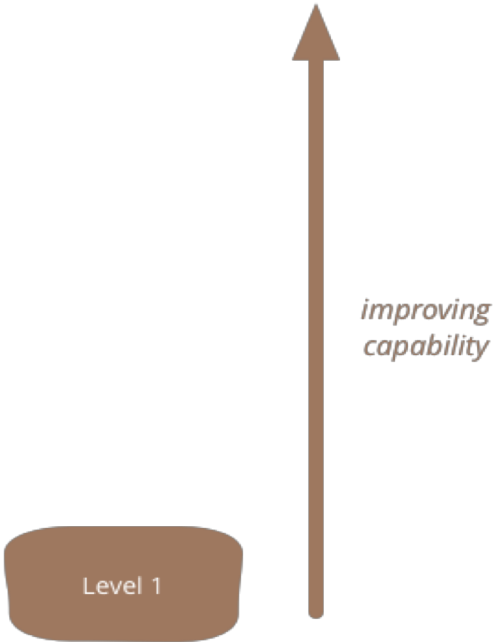
Partner with forward looking teams first

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

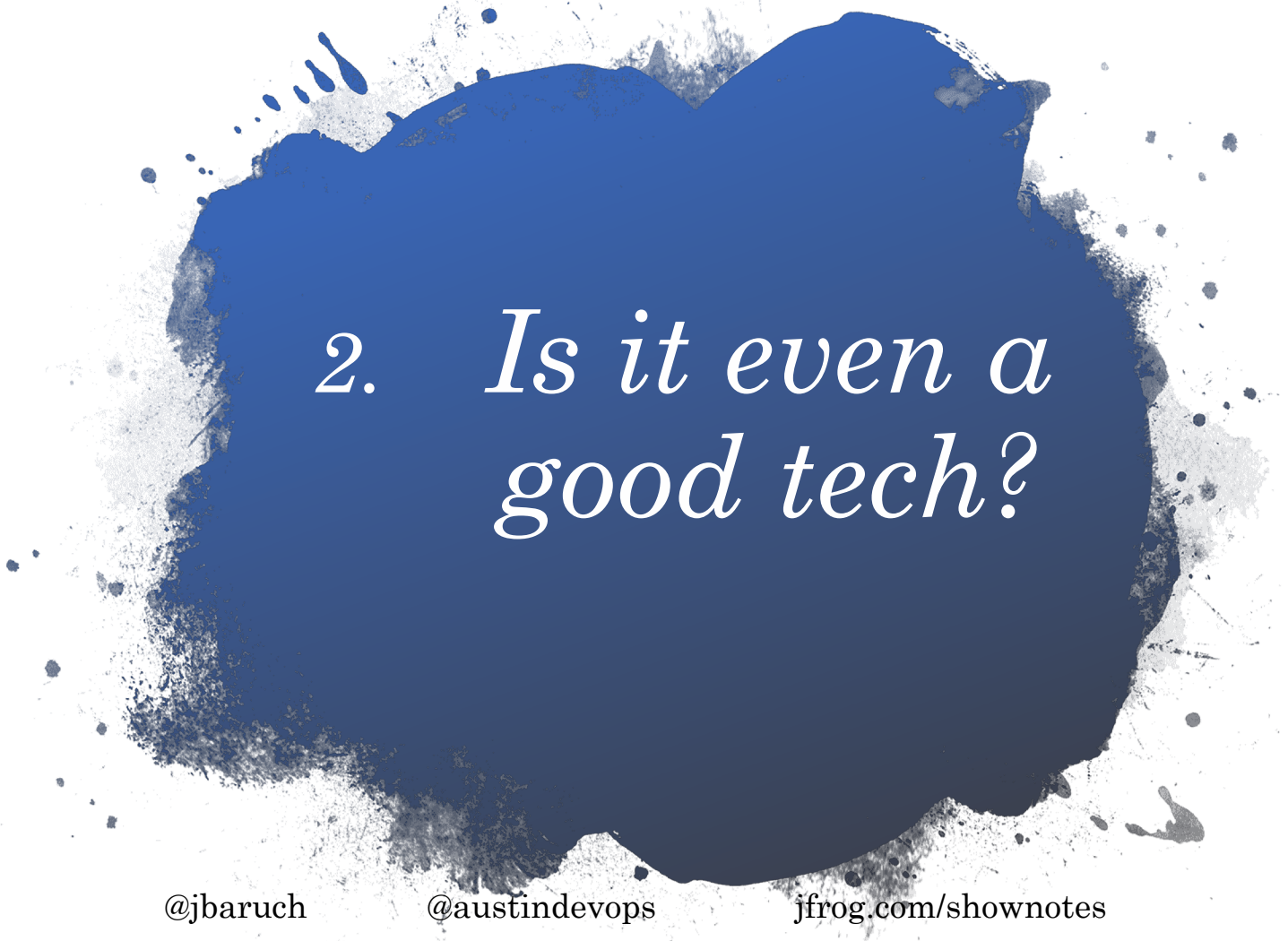
Evolve the model from time to time

And ….

Our message is:



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


2. Is it even a good tech?

FORRESTER®

IDC

Gartner

IHS Markit®

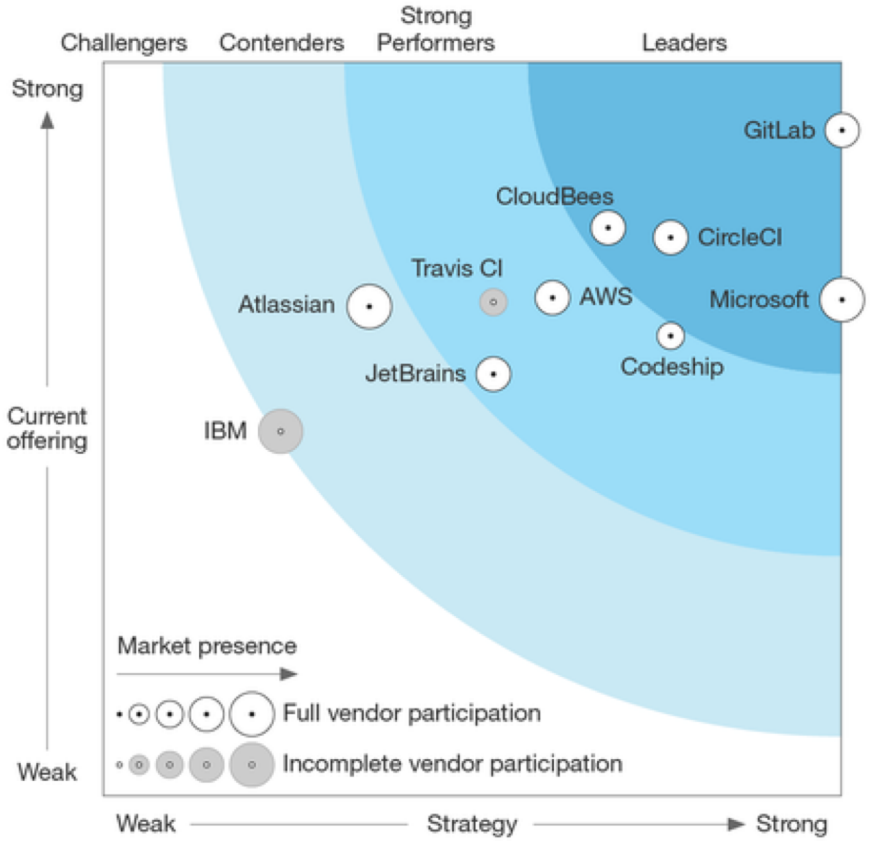
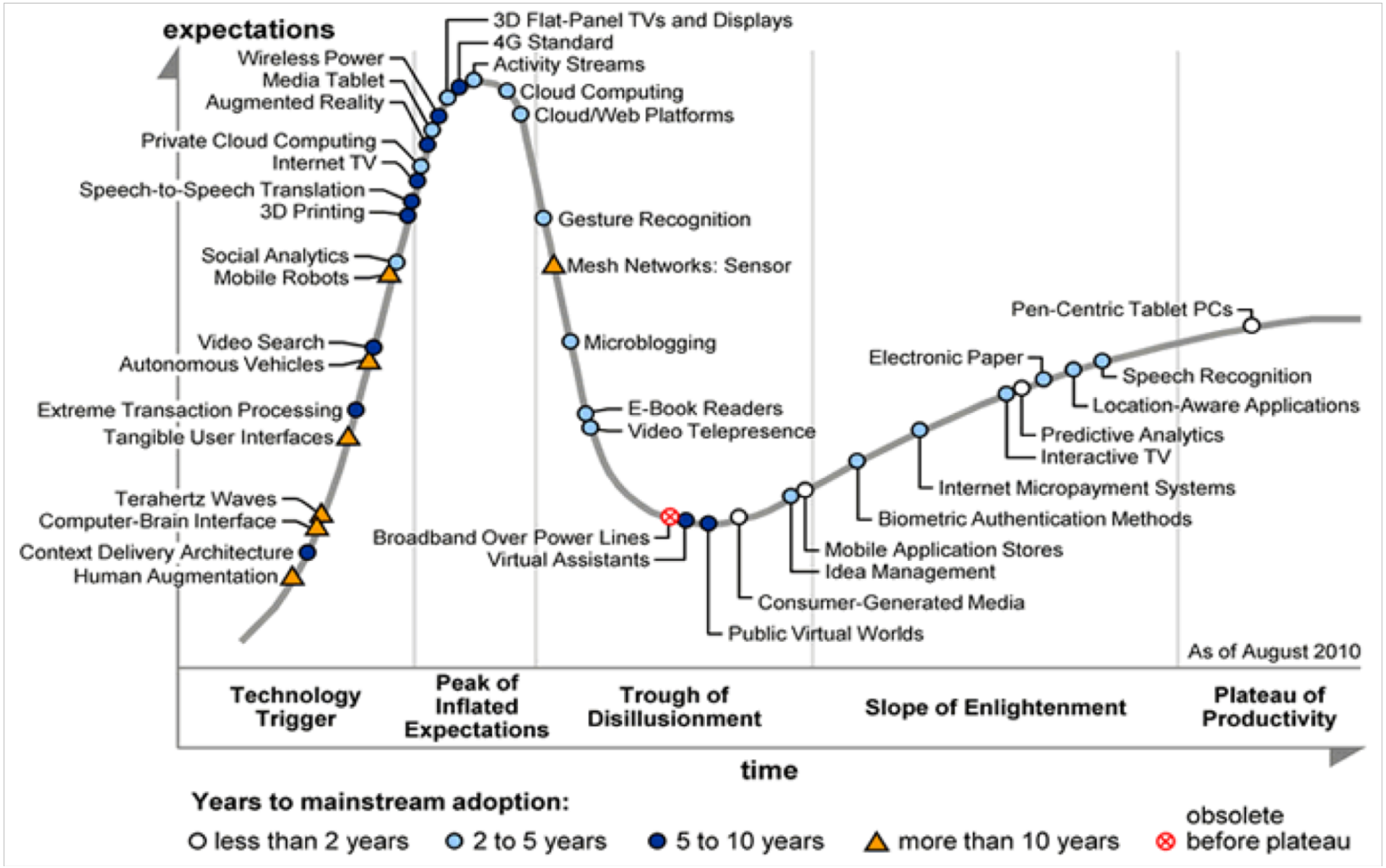
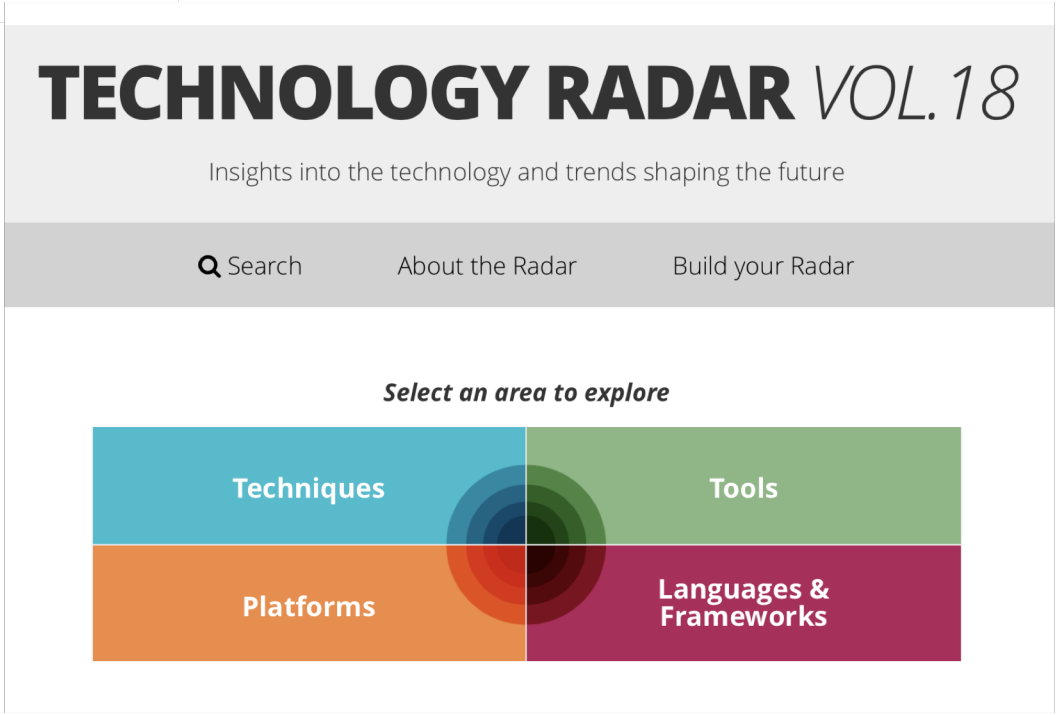

Challengers
Contenders
Strong Performers
Leaders
Strong
Current offering
Weak
GitLab
CloudBees
CircleCI
Travis CI
Atlassian
AWS
Microsoft
Codeship
JetBrains
IBM
Market presence
Full vendor participation
Incomplete vendor participation
Weak
Strategy
Strong

Figure 1. Magic Quadrant for Public Cloud Infrastructure Managed Service Providers, Worldwide
CHALLENGERS
LEADERS
NICHE PLAYERS
VISIONARIES
ABILITY TO EXECUTE
COMPLETENESS OF VISION
As of March 2017
Source: Gartner (March 2017)
Cloudreach
2nd Watch
Datapipe
Smartronix
Accenture
Melbourne IT
Logicworks
Rackspace
REAN Cloud
CSC
Bespin Global
Cognizant
Nordcloud
Claranet
Cloudnexa
Capgemini
Bulletproof
HCL Technologies
Infosys
Wipro


expectations
3D Flat-Panel TVs and Displays
4G Standard
Activity Streams
Wireless Power
Media Tablet
Augmented Reality
Cloud Computing
Cloud/Web Platforms
Private Cloud Computing
Internet TV
Speech-to-Speech Translation
3D Printing
Gesture Recognition
Social Analytics
Mobile Robots
Mesh Networks: Sensor
Pen-Centric Tablet PCs
Video Search
Autonomous Vehicles
Microblogging
Electronic Paper
Speech Recognition
Location-Aware Applications
Extreme Transaction Processing
Tangible User Interfaces
E-Book Readers
Video Telepresence
Predictive Analytics
Interactive TV
Terahertz Waves
Computer-Brain Interface
Internet Micropayment Systems
Biometric Authentication Methods
Broadband Over Power Lines
Virtual Assistants
Mobile Application Stores
Idea Management
Context Delivery Architecture
Human Augmentation
Consumer-Generated Media
Public Virtual Worlds
As of August 2010
Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Years to mainstream adoption:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau

ThoughtWorks®



@jbaruch @austindevops jfrog.com/shownotes

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?

Thank you very much!

Shownotes!
@jbaruch
@austindevops

Things that are not questions

Your résumé
Calling bullshit on the entire premise of the talk
A long rambling story with no point