

Lessons learned from 7 years of running developer platforms

platform engineering and platform as a product



OLLAS DE ALUMINIO

OLLAS DE ALUMINIO

OLLAS DE ALUMINIO

REJILLER Y CARRASAS
VICERTALES Y ANTRALES
K
COCINA

PUESTO DE INCENDIO N°

NO APTO PARA
INSTALACIONES ELECTRICAS

Deployment frequency is not good:

50% to 70% monthly or more

20% weekly or less

Hello, I am Coté

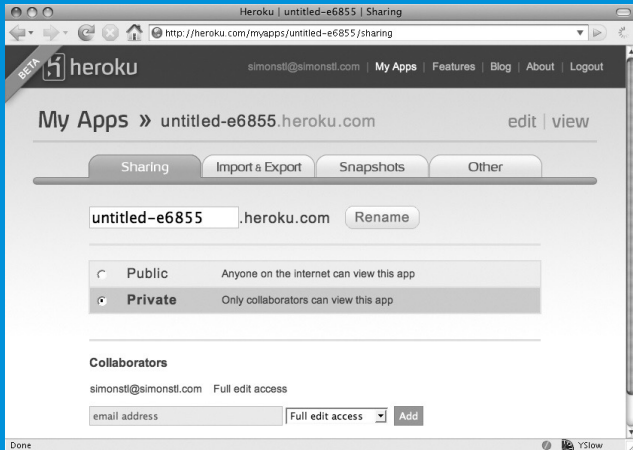
I work at **vmware**[®]

cote.io

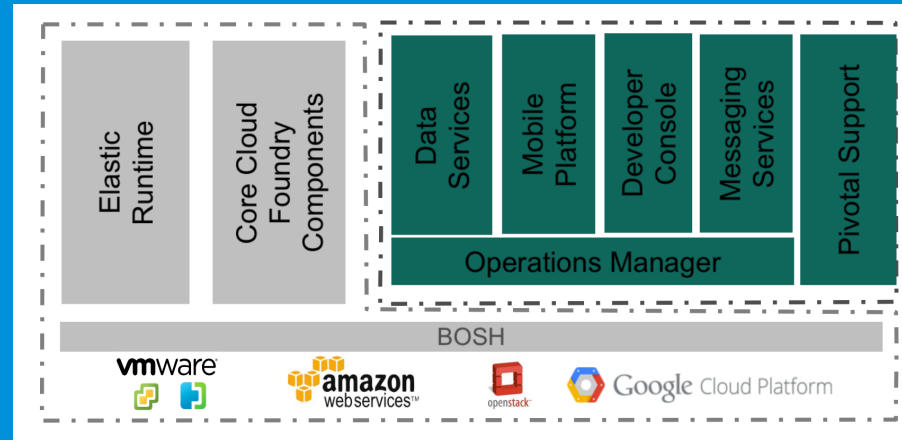


What is a platform?

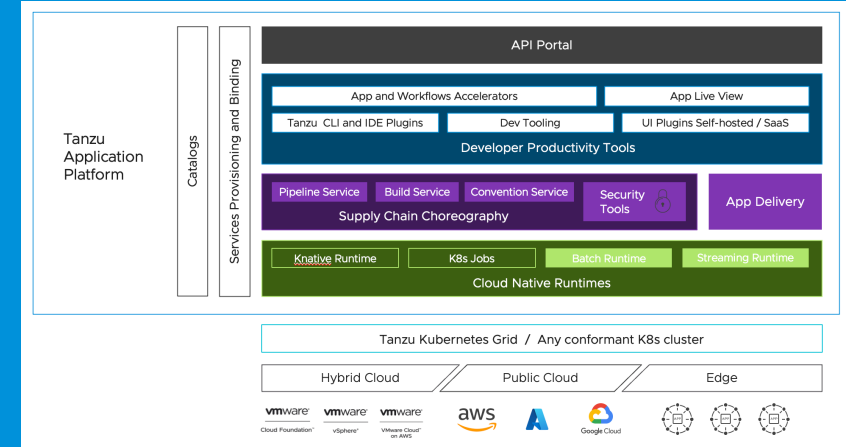
The Eternal Recurrence of (Platforms, PaaS, DevOps, Cloud Native)



2007




2015



2023 & Beyond

Not pictured:
OO, RUP, CORBA
J2EE/.Net
SOA & WS-*
Rails & LAMP
RAD, Low Code
Public Clouds

 **Kelsey Hightower** ✓
@kelseyhightower

Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.

1:04 PM - 27 Nov 2017

Last login: Thu May 24 16:34:03 on ttys001

PicklesFromTheJar:~ cote\$ █



Product (App) Teams

platform interfaces

docs and templates

portals



CLIs

APIs



INTERNAL CLOUD PLATFORM

platform capabilities

provision environments and resources

infra:
compute, network, storage

datastores:
databases, caches, buckets

messaging:
queues, brokers

identify and authenticate users and services

bind services to workloads via service discovery and secrets

support multiple tenants in one platform

store artifacts and source in registries and repos

pipelines and reconcilers for automation

enable observability via collectors and dashboards

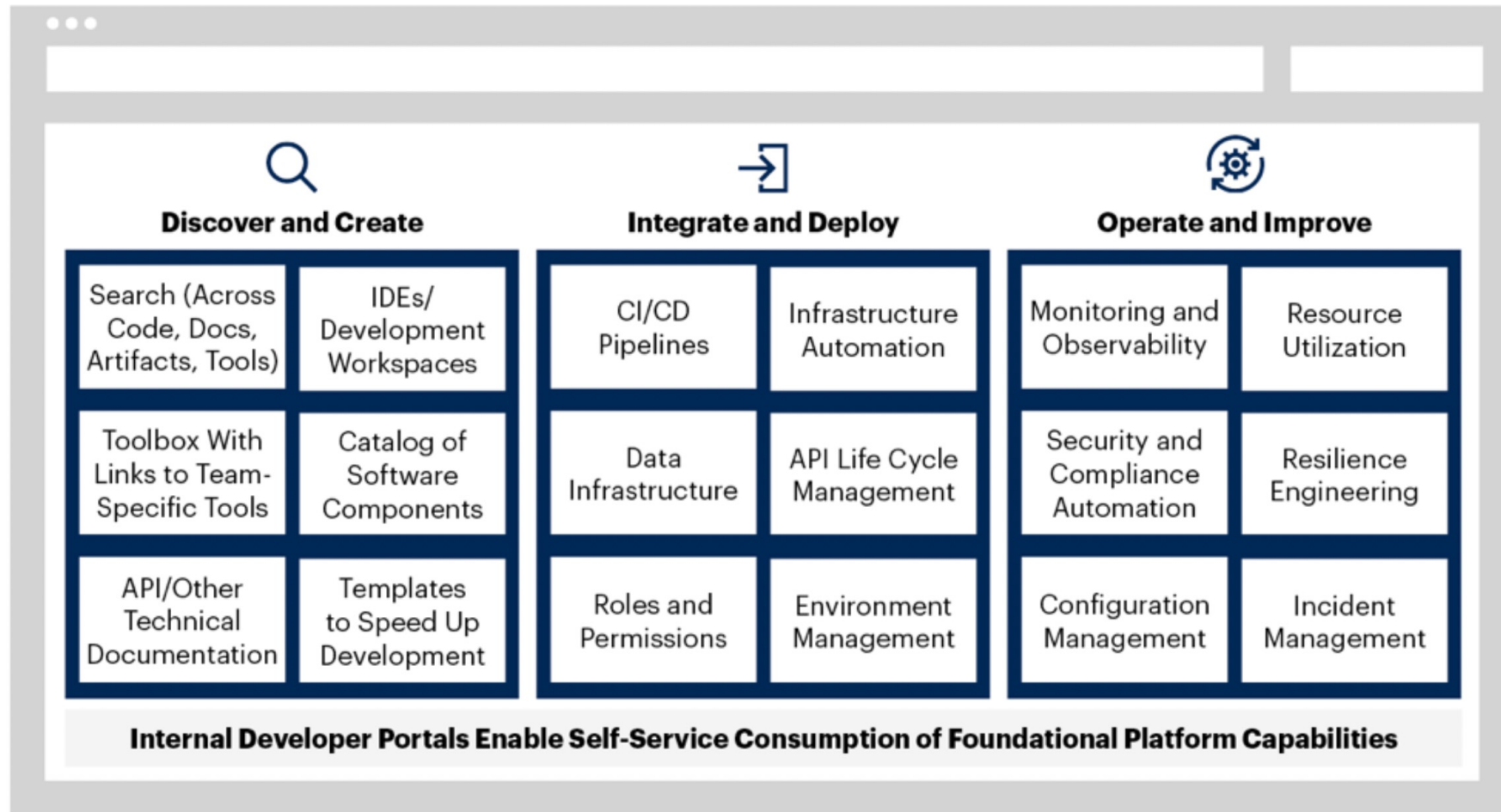
implemented by platform building blocks

Infrastructure and Operations Teams

“A digital platform is a foundation of self-service APIs, tools, services, knowledge and support which are arranged as a compelling internal product. Autonomous delivery teams can make use of the platform to deliver product features at a higher pace, with reduced coordination.”

Evan Bottcher, 2018

Internal Developer Portals Unify Disparate Platform Capabilities



Source: Gartner
761445_C

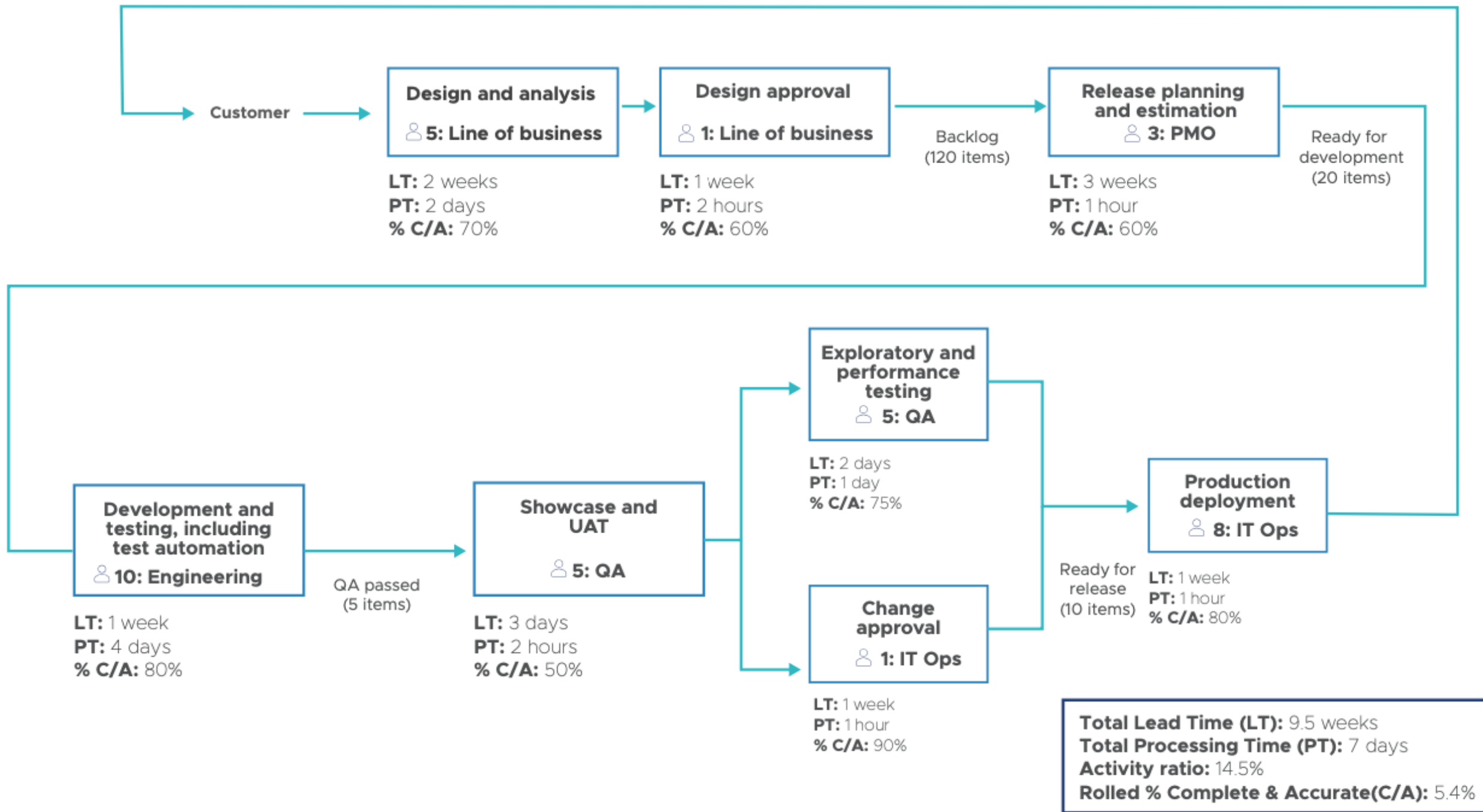
Gartner.

7 Best Practices*

Learned from

7 Years Running of Platforms

* I think I lost one of them.



1. How long does it takes to deploy one line of code?

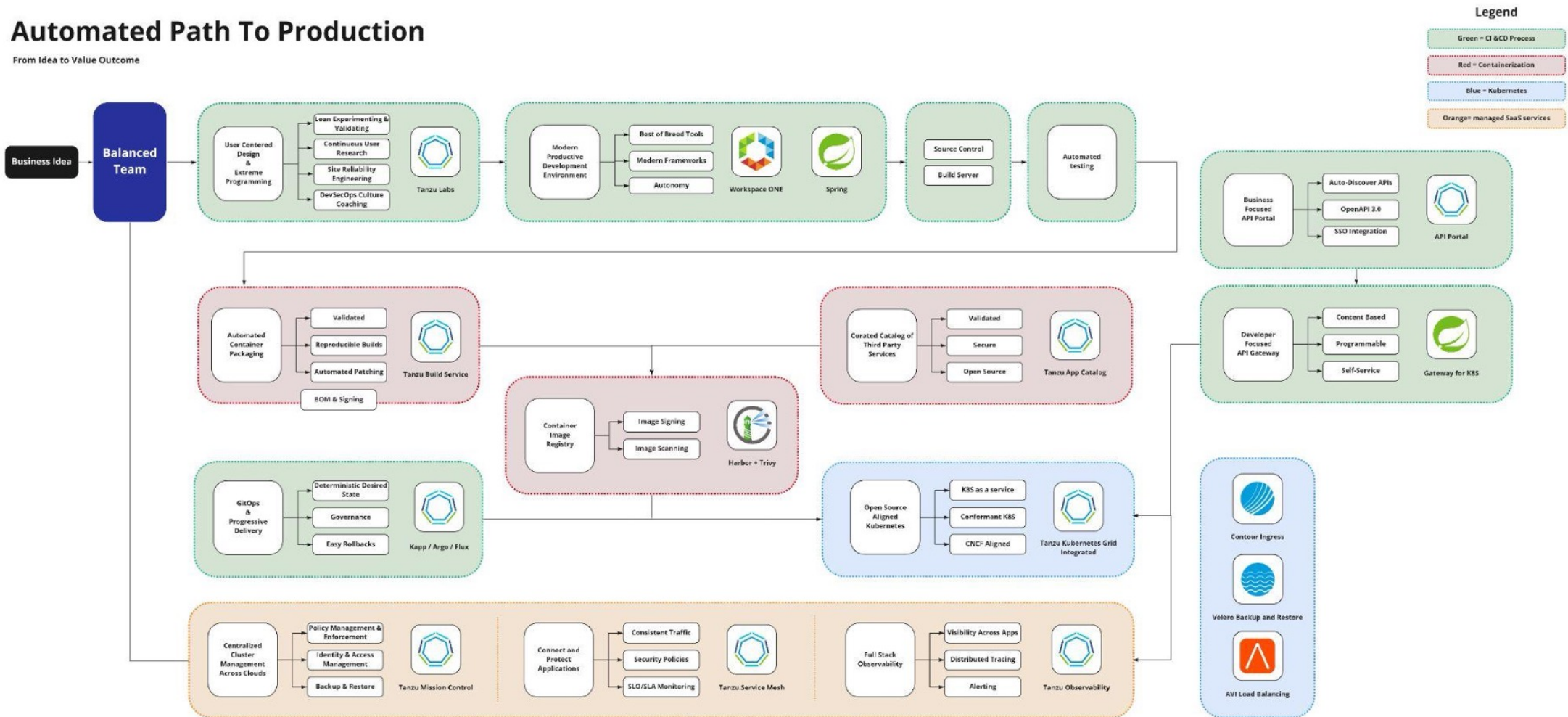
2. How long does it would take to deploy the same you have in production code over again?

Total Lead Time (LT): 9.5 weeks
Total Processing Time (PT): 7 days
Activity ratio: 14.5%
Rolled % Complete & Accurate(C/A): 5.4%

Find the golden paths and platform requirements

Automated Path To Production

From Idea to Value Outcome



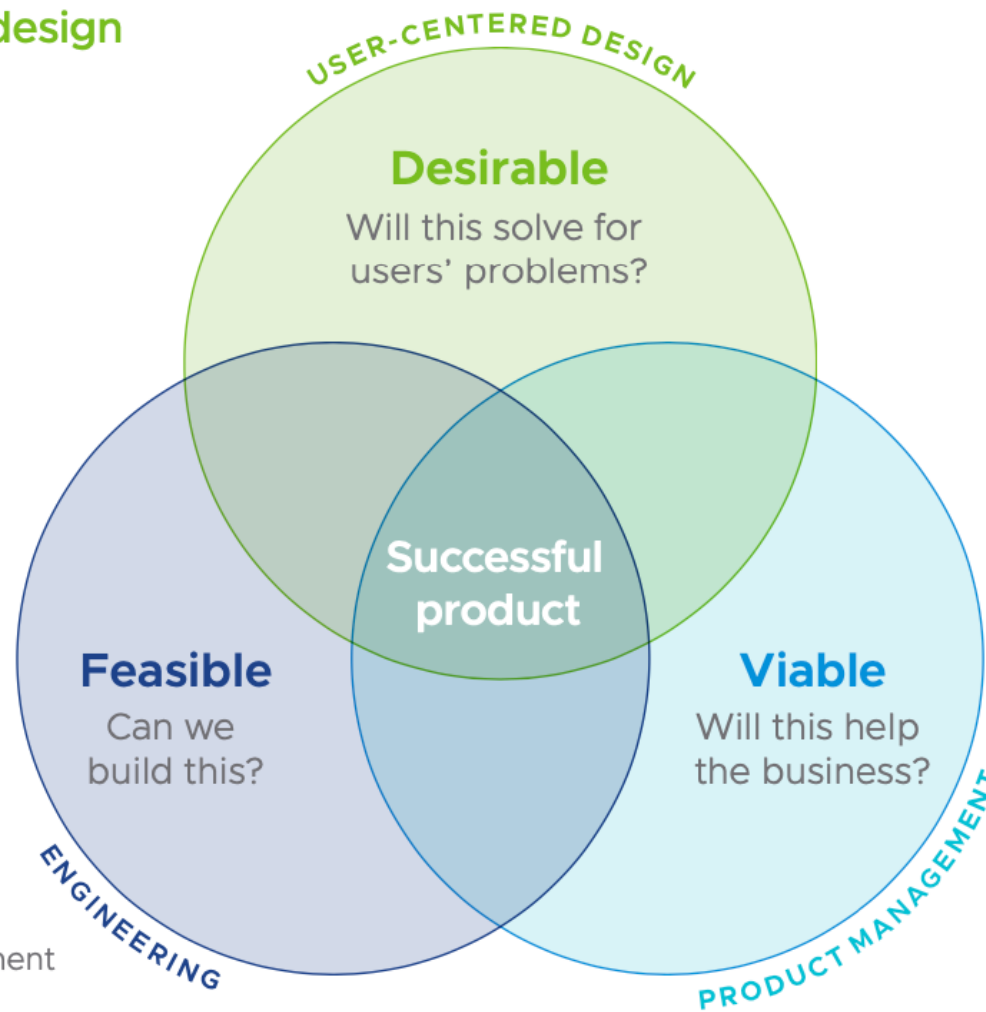
“We are building this platform not for us, we are building it for Mercedes-Benz developers.”

Thomas Müller, Mercedes-Benz

User-centered design

ACTIVITIES:

User interviews
Ethnography
Define personas
Usability testing
Service Design
UI / UX
Visual design



Agile / XP

ACTIVITIES:

Test-driven development
Pair programming
Evolutionary design
Collective code ownership
Retros
Short iterations
CI / CD

Lean Startup

ACTIVITIES:

Define product vision, strategy and roadmap
Define business model
Define minimum viable product
Identify and test assumptions
Release real product often
Understand customers
Adjust direction based on data
Constrain resources and time

Ops meet, dev – Dev, meet ops

Find the Developer Toil, Confusion, Blockers

- What are we making?
- We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I'm working.
- I am proud of the work I have delivered so far for our product.
- I am learning things that I look forward to applying to future products.
- My workstation seems to disappear out from under me while I'm working.
- It's easy to get my workstation into the state I need to develop our product.
- What aspect of our workstation setup is painful?
- It's easy to run our software on my workstation while I'm developing it.
- I can boot our software up into the state I need with minimal effort.
- What aspect of running our software locally is painful? What could we do to make it less painful?
- It's easy to run our test suites and to author new ones.
- Tests are a stable, reliable, seamless part of my workflow.
- Test failures give me the feedback I need on the code I am writing.
- What aspect of production support is painful?
- We collaborate well with the teams whose software we integrate with.
- When necessary, it is within my power to request timely changes from other teams.
- I have the resources I need to test and code confidently against other teams' integration points.
- What aspect of integrating with other teams is painful?
- I'm rarely impacted by breaking changes from other tracks of work.
- We almost always catch broken tests and code before they're merged in.
- What aspect of committing changes is painful?
- Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
- If the release process (CI/CD) fails, I'm confident something is truly wrong, and I know I'll be able to track down the problem.
- What aspect of our release process (CI/CD) is painful?
- Our team releases new versions of our software as often as the business needs us to.
- We are meeting our service-level agreements with a minimum of unplanned work.
- When something is wrong in production, we reproduce and solve the problem in a lower environment.

Platform marketing, advocacy, consulting

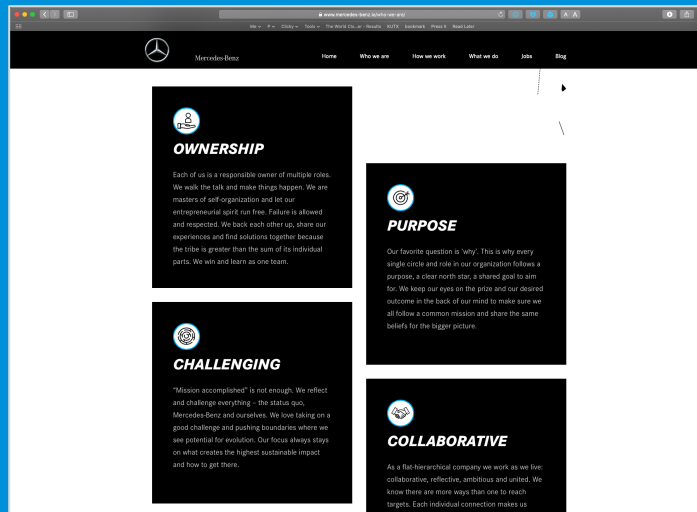
Organizational Learning



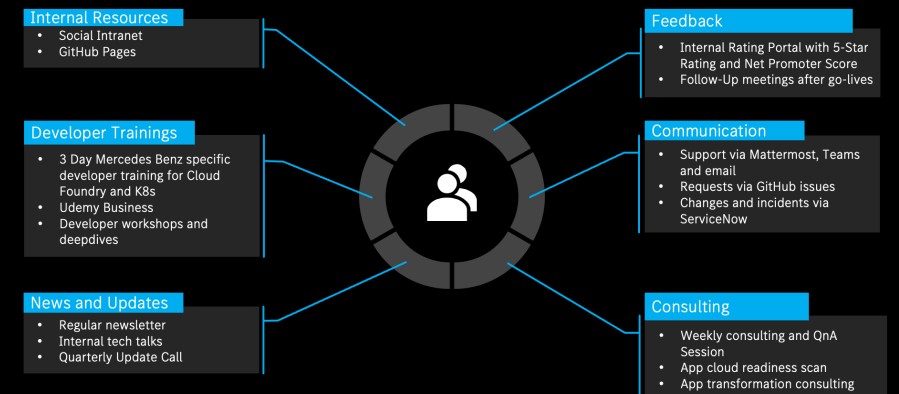
Focus on ways of working.....



BT



Developer centric platform



Mercedes-Benz

Improving Developer Productivity with Platform as a Product | Mercedes-Benz

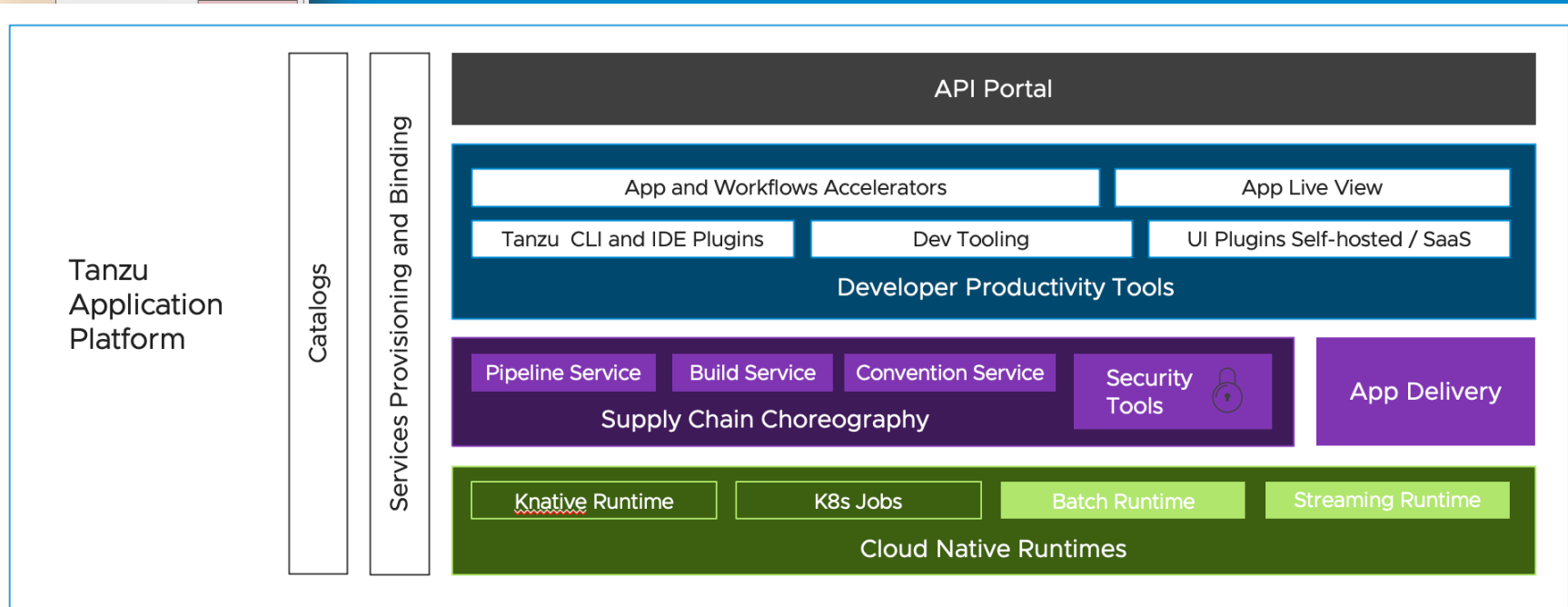
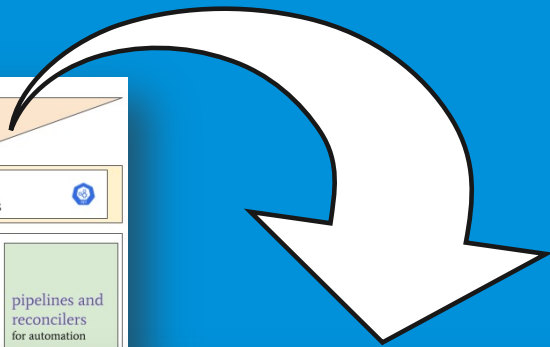
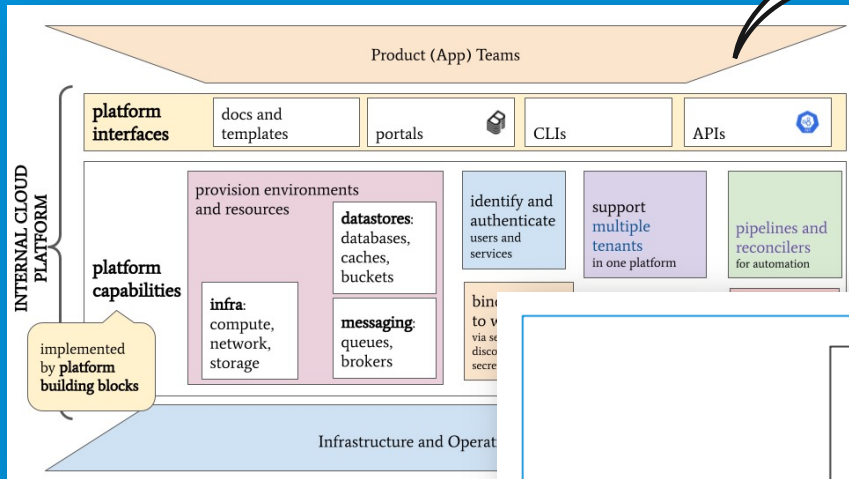
17

Start with an integrated platform, customize as find needs

- Planning - selecting from all the tools (e.g., CNCF landscape)
- Integration/Creation - making all the tools work together.
- Product Managing - Incremental updates, maintenance, and patching.
- Operating - Running the platform, diagnosing problems, security.
- Adding new features.
- Updating and maintenance.
- Support.

All of these are a full time job on their own, esp. the last two.





Tanzu Kubernetes Grid / Any conformant K8s cluster

Hybrid Cloud // Public Cloud // Edge



THE HOME DEPOT TIMELINE

2015: Handful of apps, e.g., paint desk, tool rental.

2016: ~130 apps in production.

2018: “Every week, my product and design teams are in people’s homes or [at] customer job sites, where we are bringing in a lot of real-time insights from the customers.”

2021: grew a customer’s spend from \$100k to \$300k.

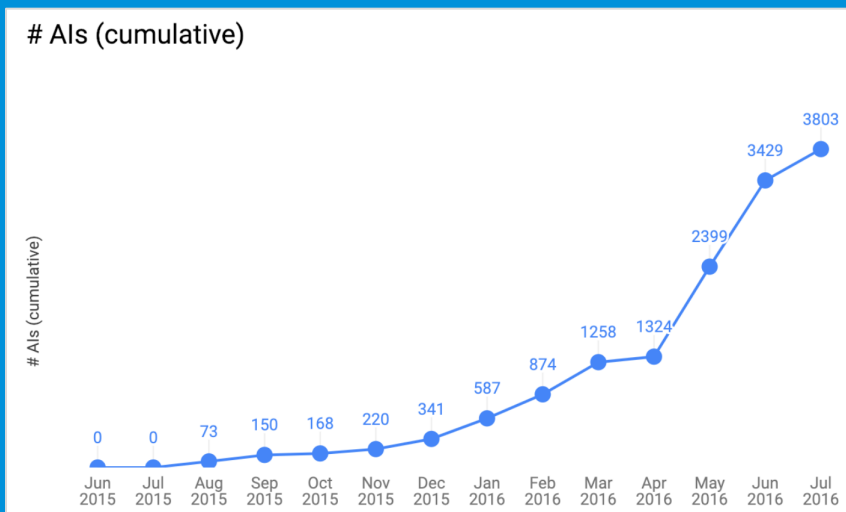
Initial Phase – Create the Platform with Developers

1. Platform team: owner, operators, engineers, advocate
2. Pick one app for business & technical feasibility.
3. Developer toil audit.
4. Find path to production with end-to-end value stream analysis
5. Start with pre-integrated platform, customize as you...
6. Build a golden path with the developer team.
7. Optionally, build an IDP as part of the platform.
8. Do this for 3 months.

Scaling Phase – Pairing & Seeding to build trust & training



1. Create platform marketing program.
2. Find two to five more apps.
3. Pair & seed from first dev & platform team to new teams.
4. "Shift Left" – build golden paths for governance, security, etc.
5. Add more infrastructure staff with pairing & seeding.
6. Do this for three months.
7. Repeat, growing number of apps as pairing & seeding allows.



Lessons learned from supporting 1,500+ application at JP Morgan Chase

A Successful Developer Experience (1/2)

1. Customer Focus: Treat internal developers like clients
2. Build, nourish and embrace a community around your platforms
3. Focus on end-to-end & deliver an integrated experience
4. Culture is critical
5. Cloud Blueprints
6. Cloud Parties
7. Self-service everything

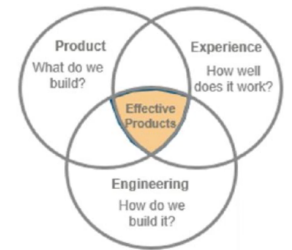


Build a customer-centre culture.
"15 Proven Techniques to Improve Customer Experience (CX)"
Blog by Snigdha Patel on the revechat.com platform

J P M o r g a n C h a s e

A Successful Developer Experience (2/2)

8. Clear responsibility model, boundaries and platform contract
9. Operationally stable, reliable, and has well-defined SLOs
10. Inherently secure
11. Streamline tooling for CI/CD
12. Enable innovation through managing risk
13. Automate, automate, automate!
14. Short time to Hello World!
15. Partner for success



J P M o r g a n C h a s e

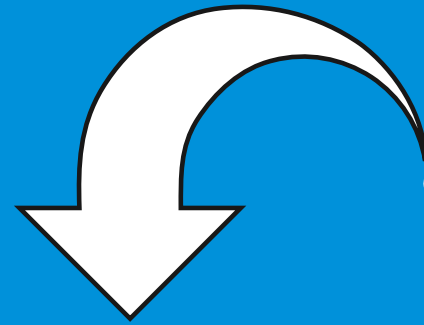
Thanks!

 <https://newsletter.cote.io>

 <https://cote.io/platform/>

 cotem@vmware.com

 [@cote@hachyderm.io](https://twitter.com/cote)



Slides & stuff

