

Lessons Learned from Running Developer Platforms & Communities

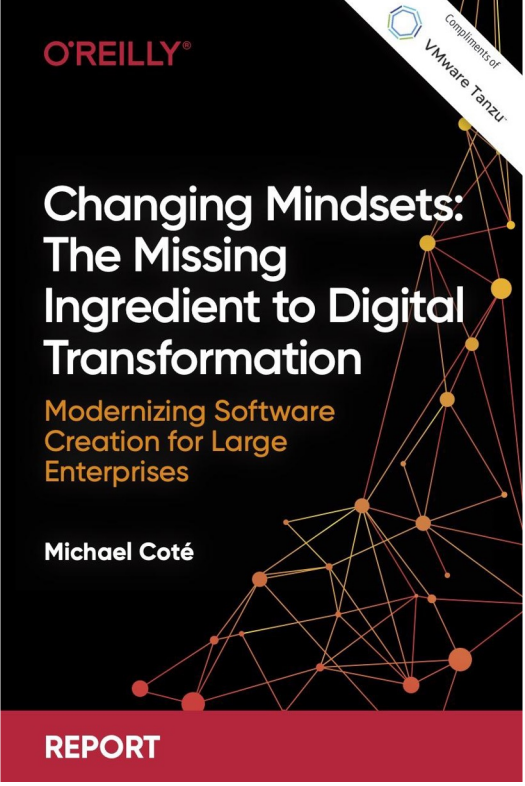
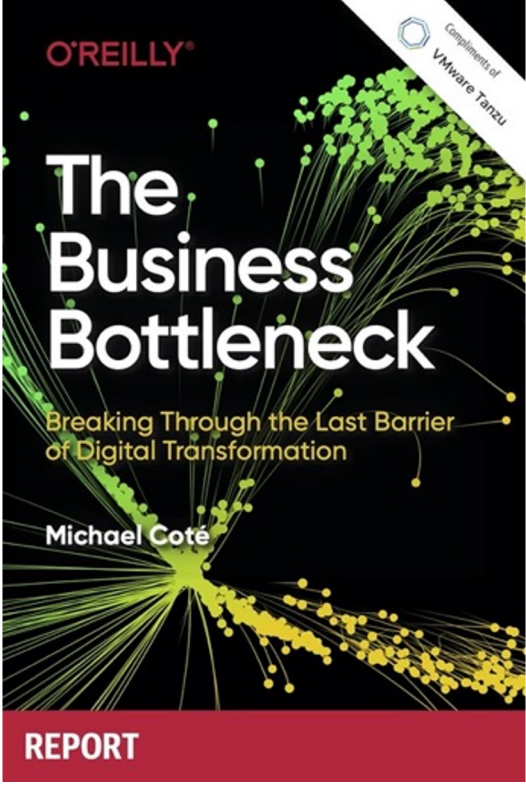
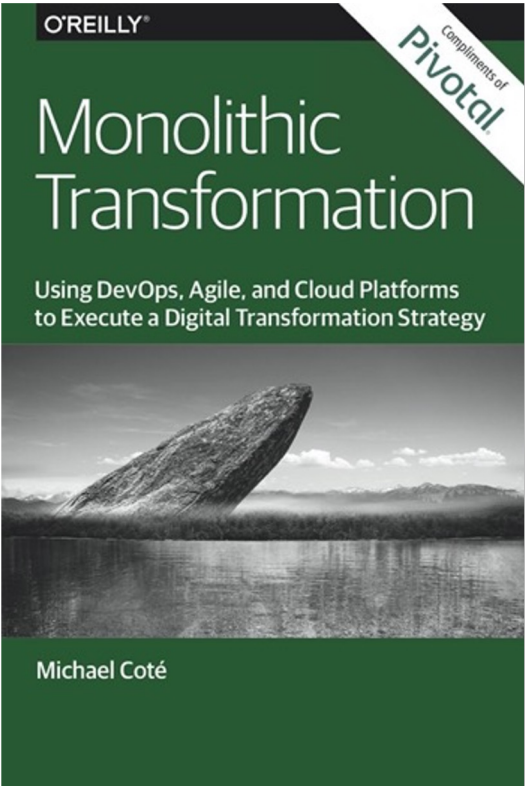
Platform engineering, platform as a
product

Coté

November 11th, 2022

Coté

@cote | <http://cote.io> | cotem@vmware.com



There's more interest than ever in "platform engineering"

Mark O'Neill
@TheMarkONeill

The Gartner Hype Cycle for Software Engineering 2022 is published! Notable changes this year include Machine Learning Code Generation entering the Hype Cycle, many technologies at the Peak, and Microservices finally escaping the Trough of Disillusionment

<https://gartner.com/interactive/hc/4017202>

2:56 AM · Aug 11, 2022 from Boston, MA · Twitter Web App

The Dream:

Development
learns the best
problems to
solve, how to
solve them, and
then keeps
doing it

Source: Pivotal Conversations #113, Sep 2018. Pic: Tyler Olson.

An exaggerated tale..



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

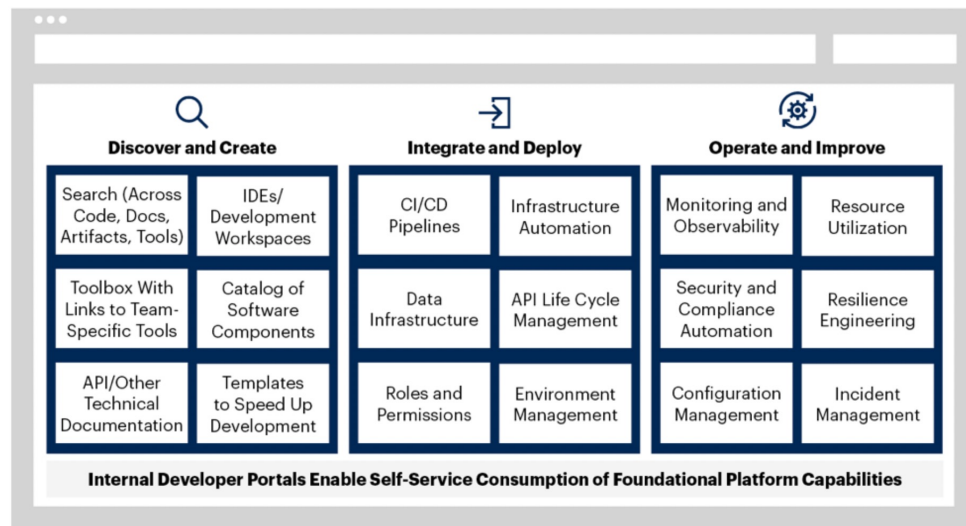
PicklesFromTheJar:~ cote\$ █



Platforms == Internal Developer Portals + CaaS

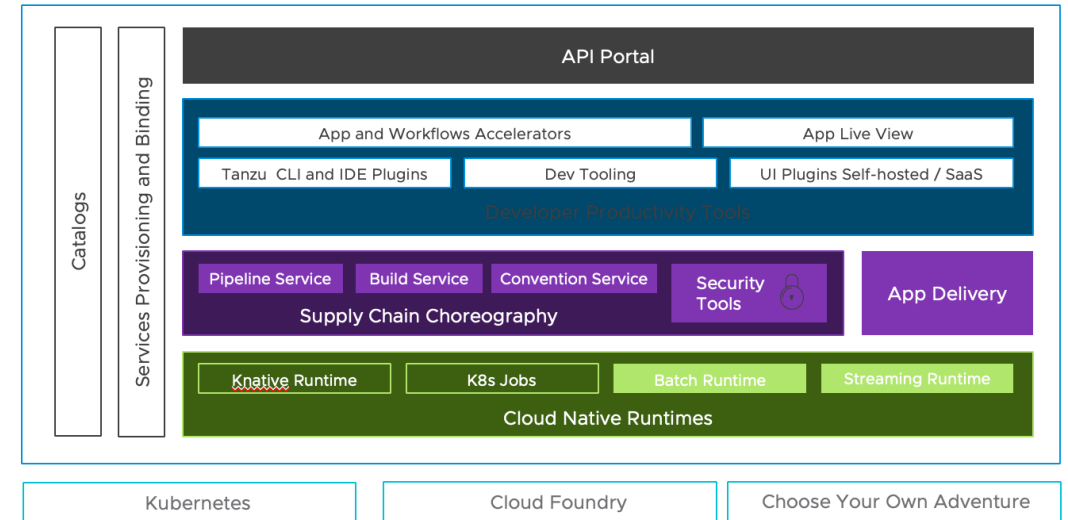
...more or less...

Internal Developer Portals Unify Disparate Platform Capabilities



Source: Gartner
761445_C

Gartner's IDP model



VMware Tanzu Platform Model

The most important part: product manage the platform

Ops people treating developers as customers



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



Thomas Müller,
Mercedes Benz.

Platform Engineering and Advocacy at JP Morgan Chase

Gaia Kubernetes Platform Speeds and Feeds

>100
GLOBAL
CLUSTERS



>700 TB
Memory
Capacity
Deployed

>4000
SERVERS
SUPPORTED

>345
PRODUCTION
APPS

~300
OS REPAVES
PER DAY

JPMorgan Chase

Gaia Application Platform Speeds and Feeds

>1522
Production
Applications



>206k
Container
Workloads

>331k
Running Memory

37 Pools
16 Data Centers
/ 3 regions

~15k
Deployments
/Day

JPMorgan Chase

Source: ["Improving JPMorgan Chase's Developer Experience on the Cloud,"](#) Nadi Away, JPMC, June 2022.

Building Your Platform Engineering Strategy



Platform as a Product

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

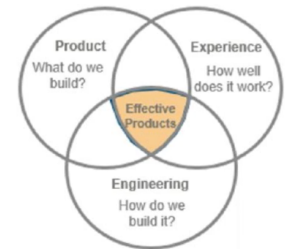


Build a customer-centre culture.
"15 Proven Techniques to Improve Customer Experience (CX)"
Blog by Snigdha Patel on the revechat.com platform

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

Source: ["Improving JPMorgan Chase's Developer Experience on the Cloud,"](#) Nadi Away, JPMC, June 2022.

Application developers build application, not platforms

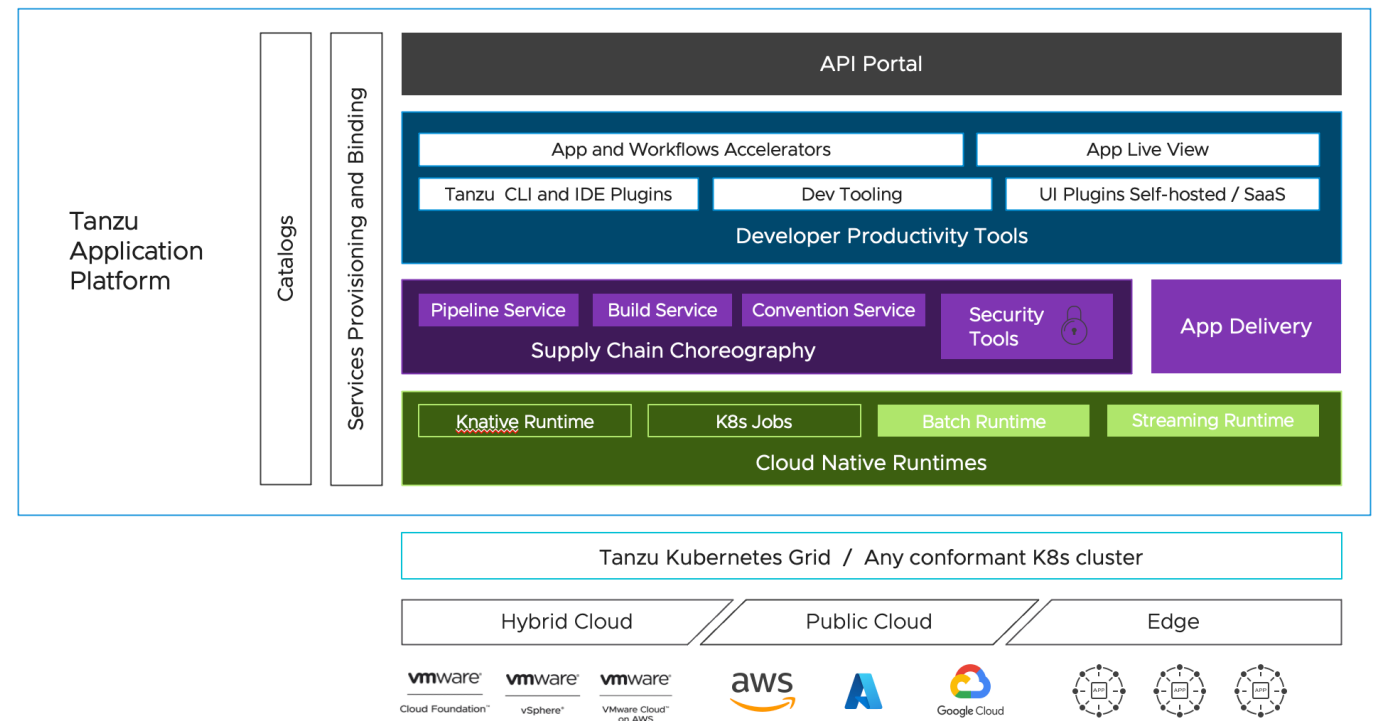
- Planning - selecting from all the tools (e.g., CNCF landscape)
- Integration/Creation - making all the tools work together.
- Product Managing - Incremental updates, maintenance, and patching.
- Operating - Running the platform, diagnosing problems, security.
- Funding - who pays for it? Right before the check comes, developers always end up going to the toilet.

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



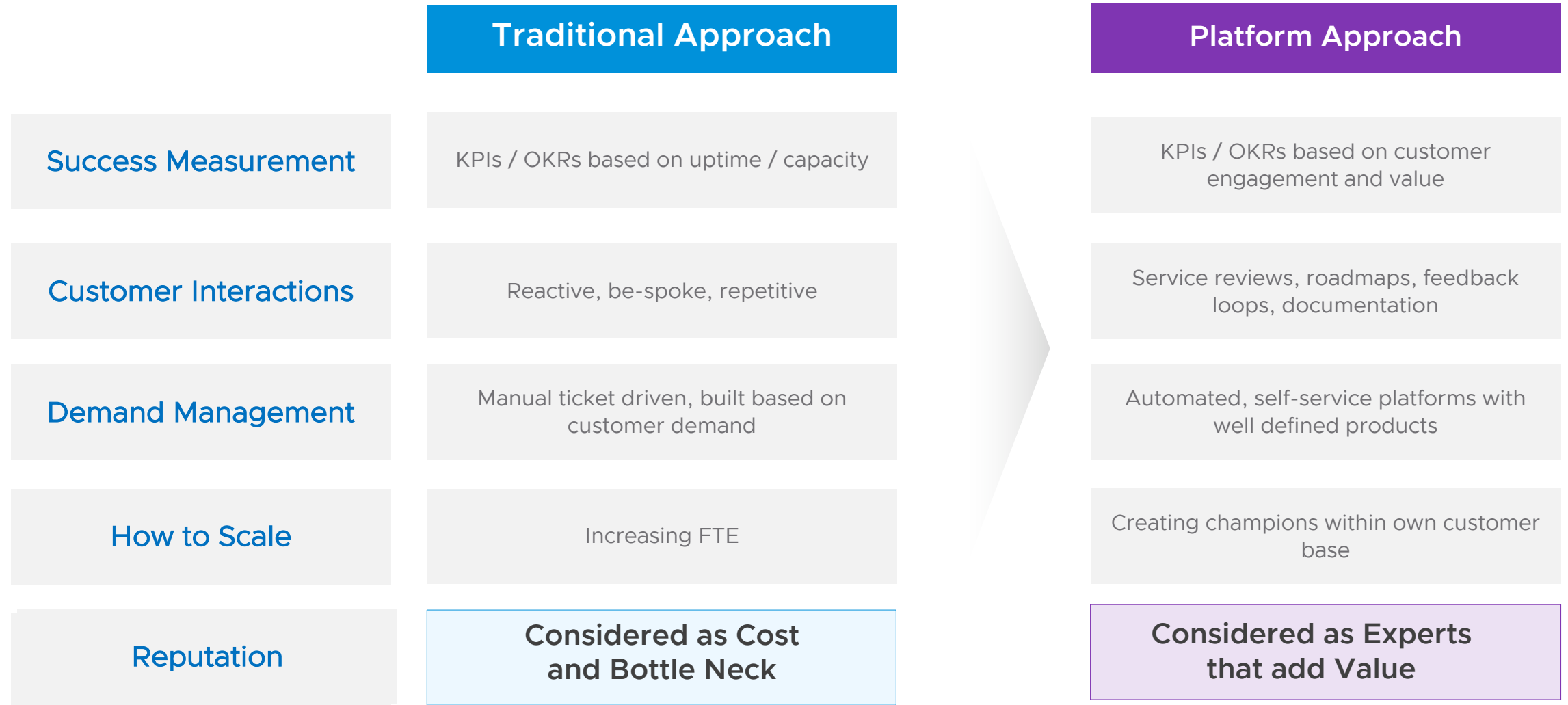
Avoid Building Your Own Platform, Start with Existing Platforms & Components

- There are many, many options.
- Tanzu Application Platform
- Tanzu Application Service.
- PaaSes like Cloud Foundry.

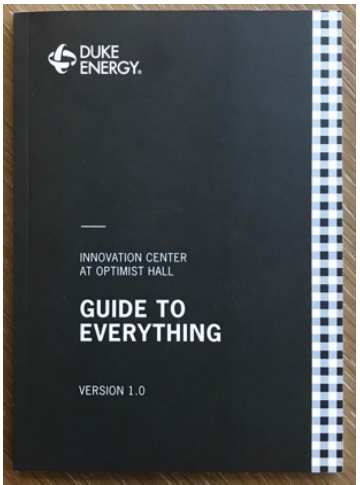
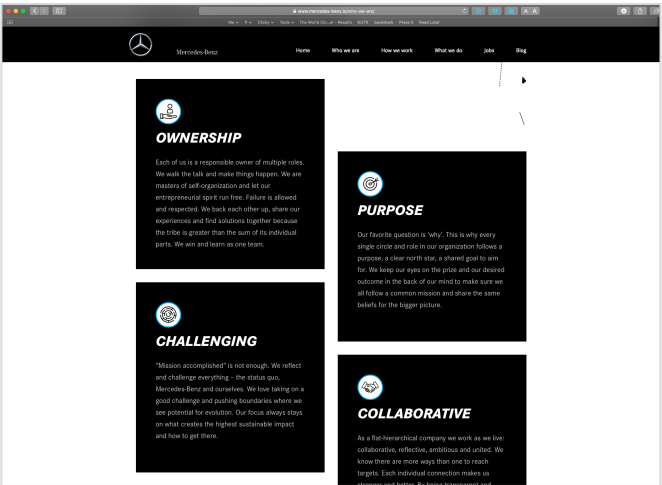


Sources: [Knative Fundamentals](#), Jacques Chester, 2020; ["Recognizing and Removing Friction Points in the Developer Experience on Kubernetes,"](#) Nivedita Ghosh, Nov, 2021.

Transform the ops culture

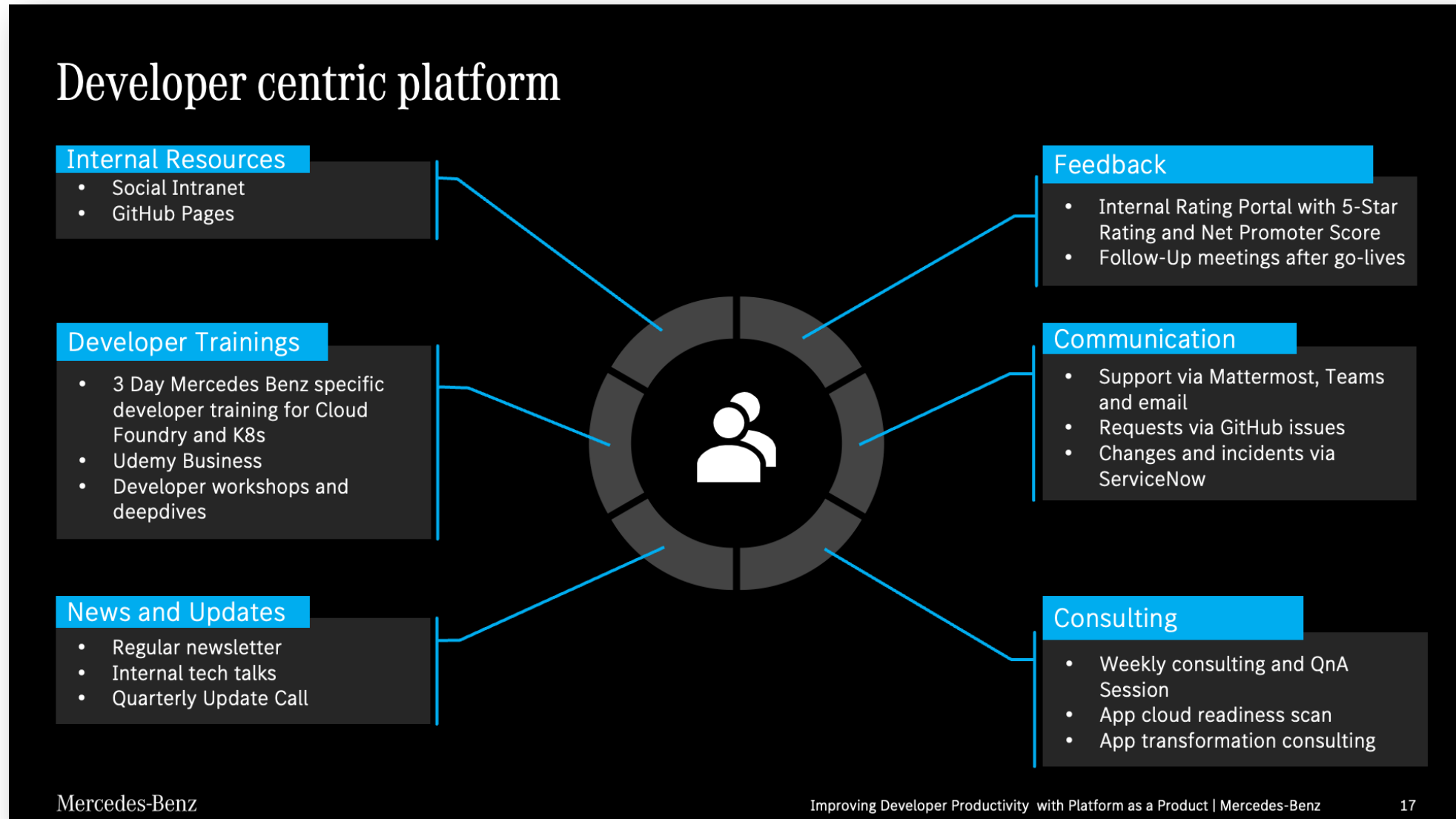


Drive Platform Usage & Practices with Internal Marketing, Consulting, Branding



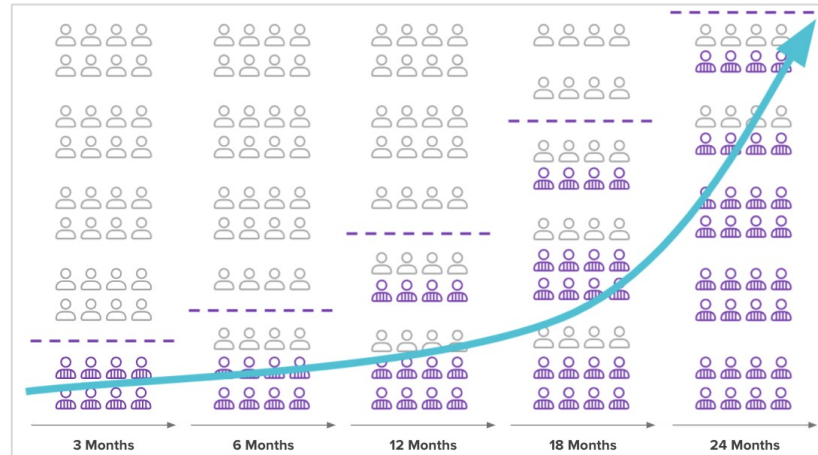
Sources: BT Canvas team; [MB.io](#); Talanx; Duke Energy; Allstate; "Take DevOps to 11 and Sprinkle Cloud on it with Rainbows and Unicorns," Matt Curry, s1p 2017; customer discussions.

Platform advocacy, developer relations, consulting



Getting started

Starting Small, Seeding & Winning Trust



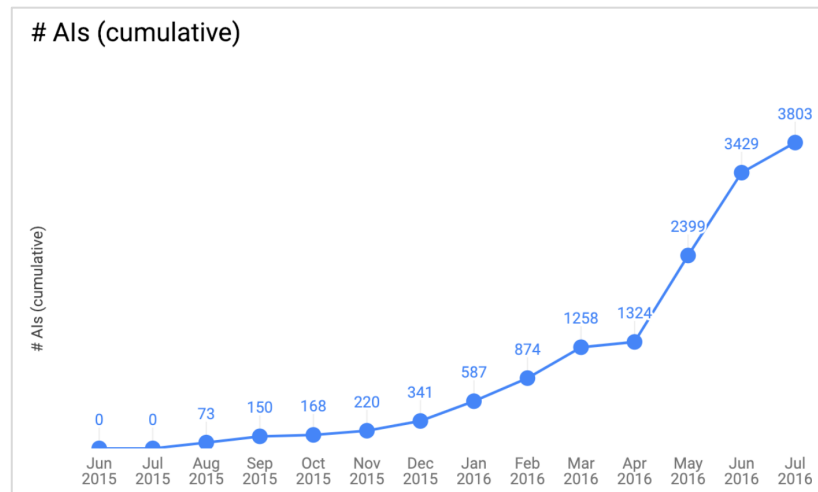
Home Depot Timeline

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: one customer’s spend from \$100k to \$300k.

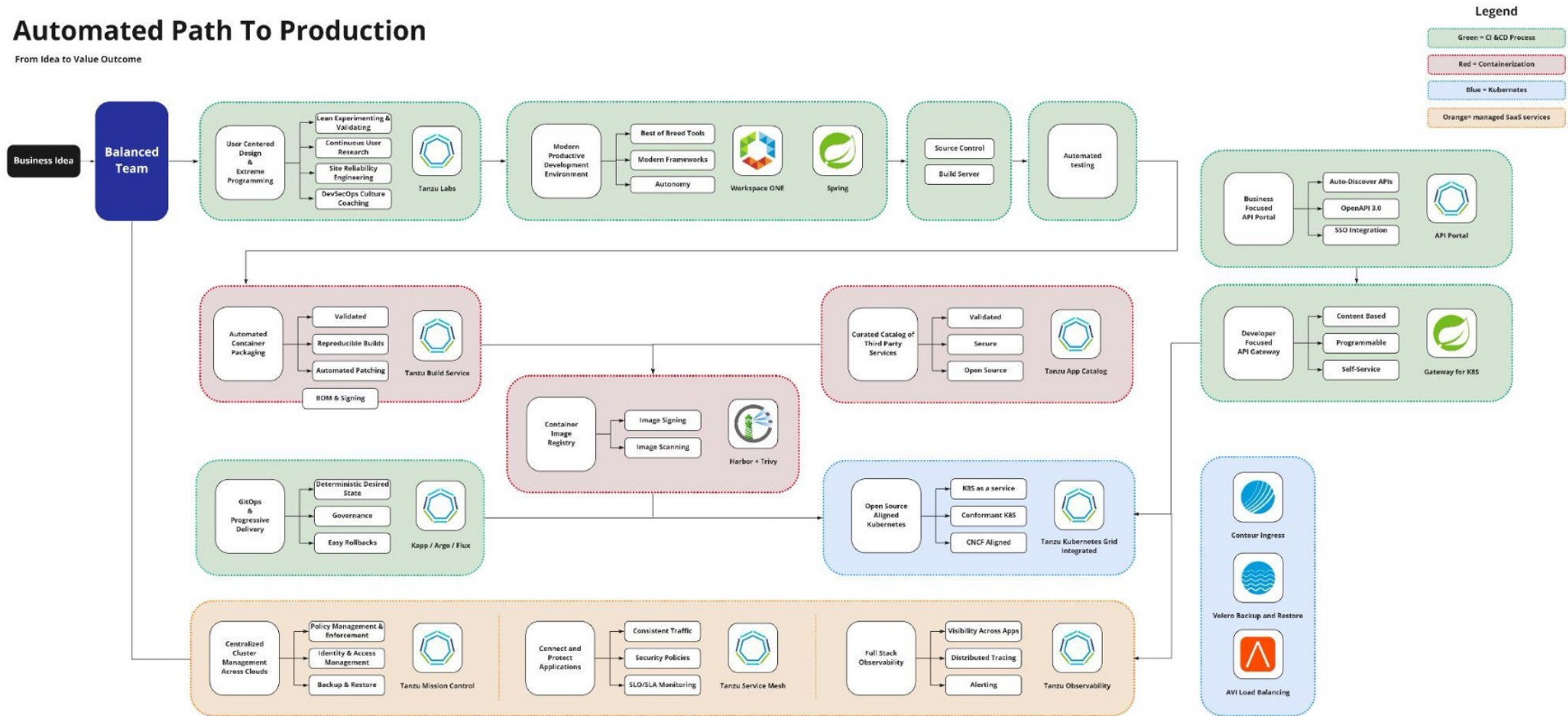


Sources: “From 0 to 1000 Apps: The First Year of Cloud Foundry at The Home Depot,” Anthony McCulley, The Home Depot, Aug 2016; “Cloud Native at The Home Depot, with Tony McCulley,” Pivotal Conversations #45; USAF presentations and write-ups; “Driving Business Agility Without Large-Scale Transformation Programs,” Venkatesh Arunachalam, Sep 2021; *The Business Bottleneck*, Coté.

Find the golden paths and platform requirements

Automated Path To Production

From Idea to Value Outcome



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.

Sources: ["Tech Debt Audit: How to Prioritize and Reduce the Tech Debt that Matters Most,"](#) Susie Forbath and Tyson McNulty, VMware Tanzu, 2022.

Learn More!

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

Email cotem@vmware.com for slides.

