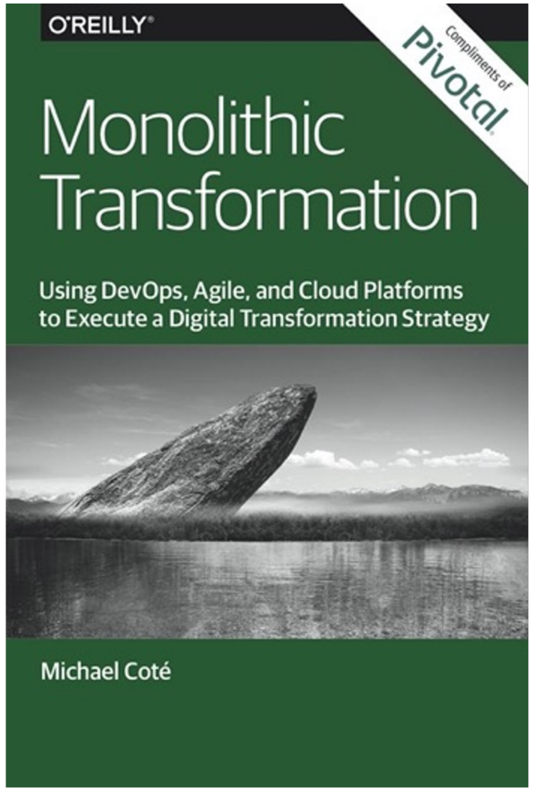
Surviving the platform journey

Coté - March 28th, 2024

Coté

https://newsletter.cote.io/ | cote@broadcom.com

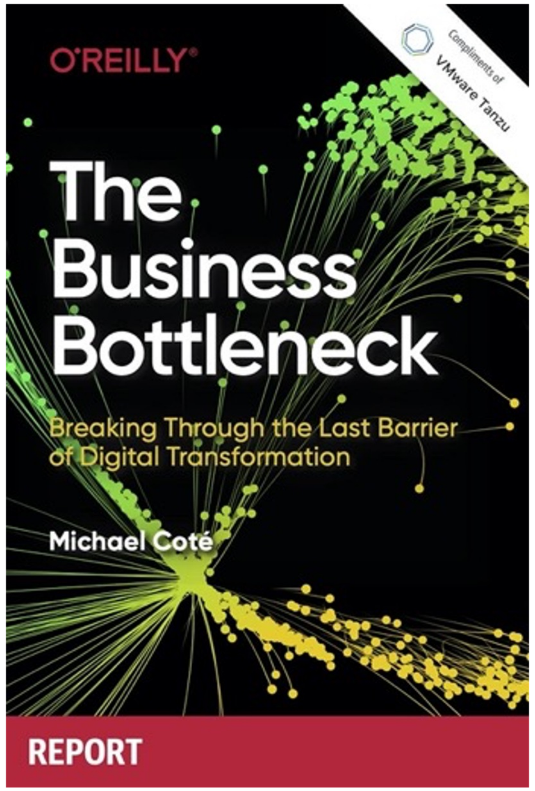

O'REILLY®

Compliments of Pivotal

Monolithic Transformation

Using DevOps, Agile, and Cloud Platforms to Execute a Digital Transformation Strategy

Michael Coté


O'REILLY®

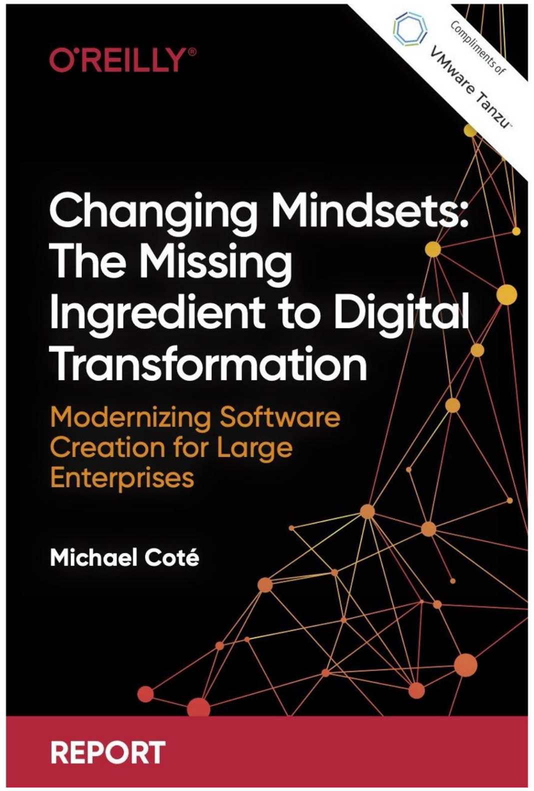
Compliments of VMware Tanzu

The Business Bottleneck

Breaking Through the Last Barrier of Digital Transformation

Michael Coté

REPORT


O'REILLY®

Compliments of VMware Tanzu

Changing Mindsets: The Missing Ingredient to Digital Transformation

Modernizing Software Creation for Large Enterprises

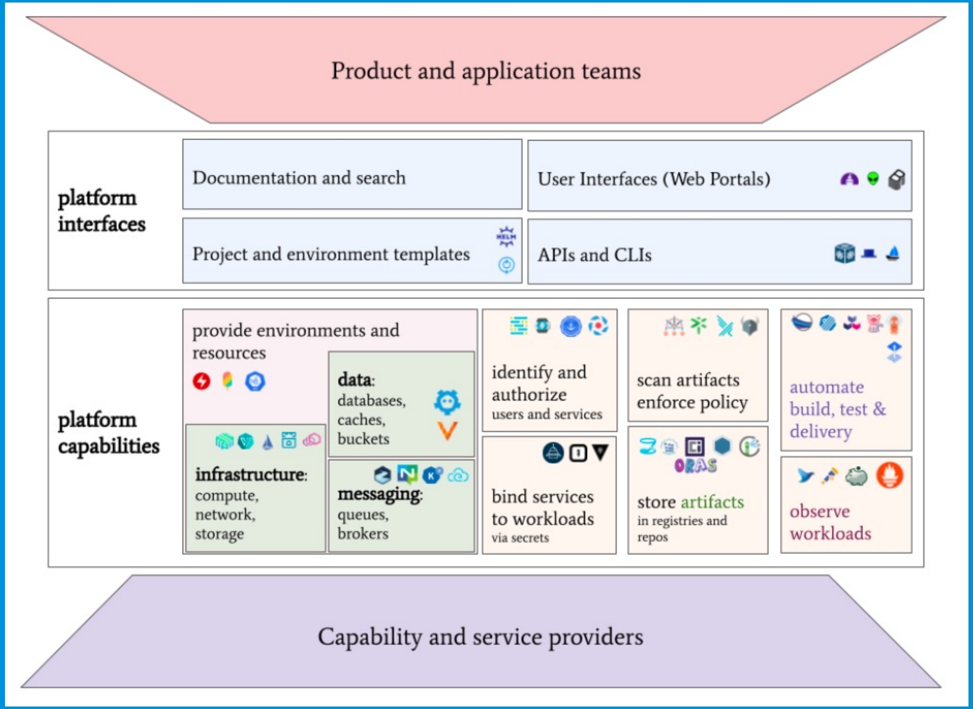
Michael Coté

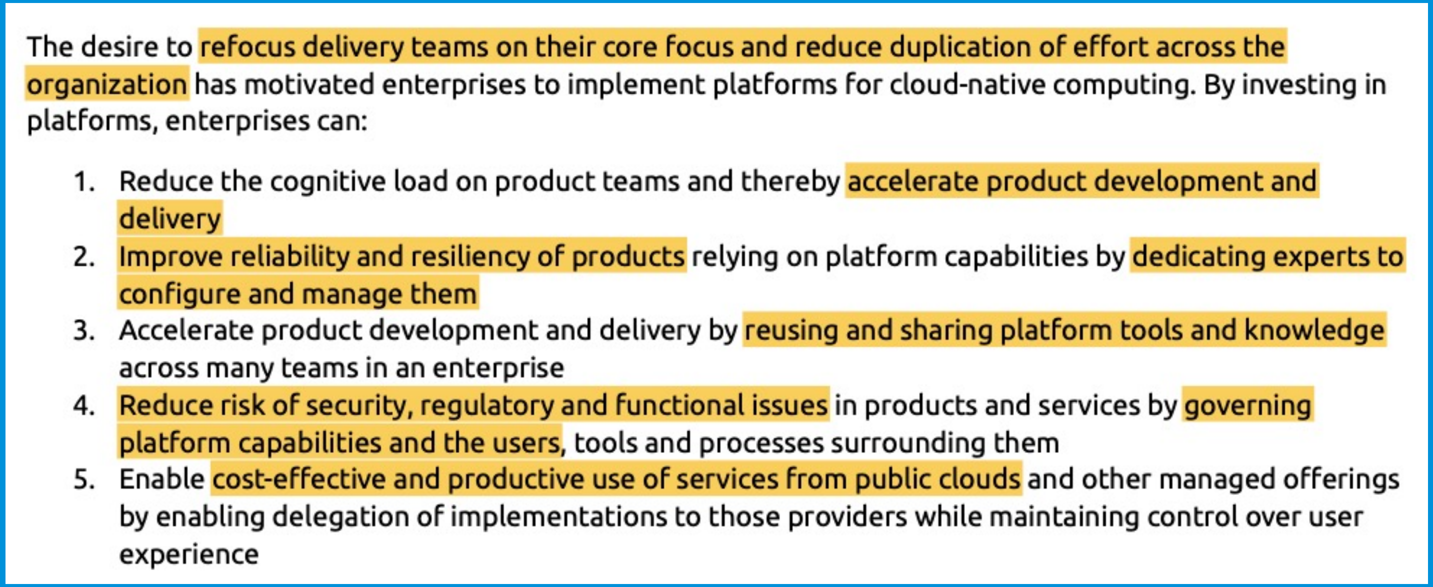
REPORT

Tanzu Talk

SOFTWARE DEFINED TALK

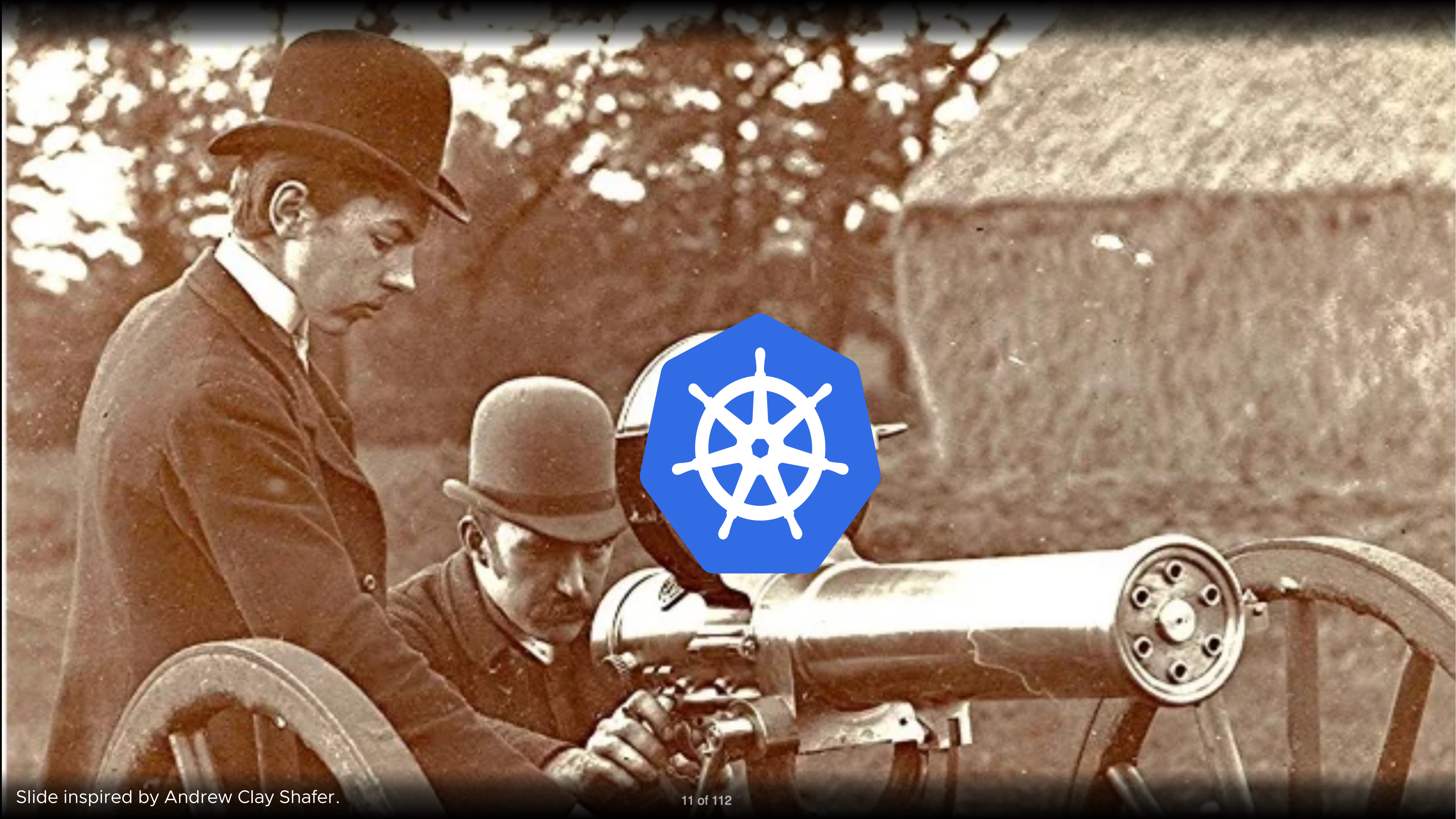
RedMonk | DELL | 451 Research® | Pivotal | vmware® | Tanzu® by Broadcom


Product and application teams
platform interfaces
Documentation and search
User Interfaces (Web Portals)
Project and environment templates
APIs and CLIs
platform capabilities
provide environments and resources
data: databases, caches, buckets
infrastructure: compute, network, storage
messaging: queues, brokers
identify and authorize users and services
bind services to workloads via secrets
scan artifacts enforce policy
store artifacts in registries and repos
automate build, test & delivery
observe workloads
Capability and service providers


The desire to refocus delivery teams on their core focus and reduce duplication of effort across the organization has motivated enterprises to implement platforms for cloud-native computing. By investing in platforms, enterprises can:

1. Reduce the cognitive load on product teams and thereby accelerate product development and delivery
2. Improve reliability and resiliency of products relying on platform capabilities by dedicating experts to configure and manage them
3. Accelerate product development and delivery by reusing and sharing platform tools and knowledge across many teams in an enterprise
4. Reduce risk of security, regulatory and functional issues in products and services by governing platform capabilities and the users, tools and processes surrounding them
5. Enable cost-effective and productive use of services from public clouds and other managed offerings by enabling delegation of implementations to those providers while maintaining control over user experience

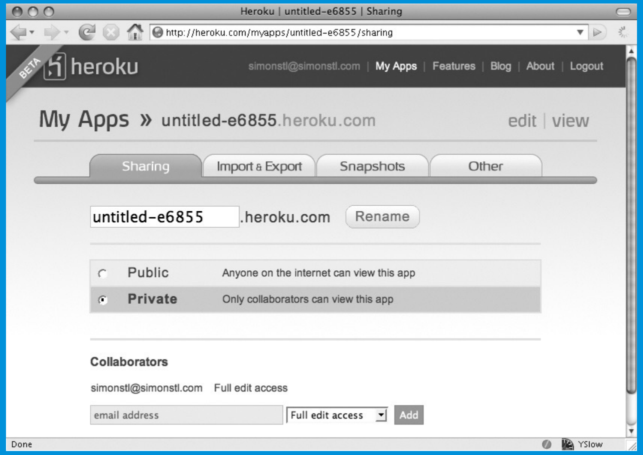
Source: "CNCF Platforms White Paper," March 2023.



Slide inspired by Andrew Clay Shafer.

The Eternal Recurrence of (Platforms, PaaS, DevOps, Cloud Native, Golden Paths, Platform Engineering, . . .)

(╯°□°) ╯︵ ┻━┻)   (╯°□°) ╯︵ ┻━┻)



Heroku | untitled-e6855 | Sharing
http://heroku.com/myapps/untitled-e6855/sharing
BETA
heroku
simonstl@simonstl.com | My Apps | Features | Blog | About | Logout
My Apps » untitled-e6855.heroku.com
edit | view
Sharing
Import & Export
Snapshots
Other
Rename
Public
Private
Collaborators
email address
Add
Done
YSlow

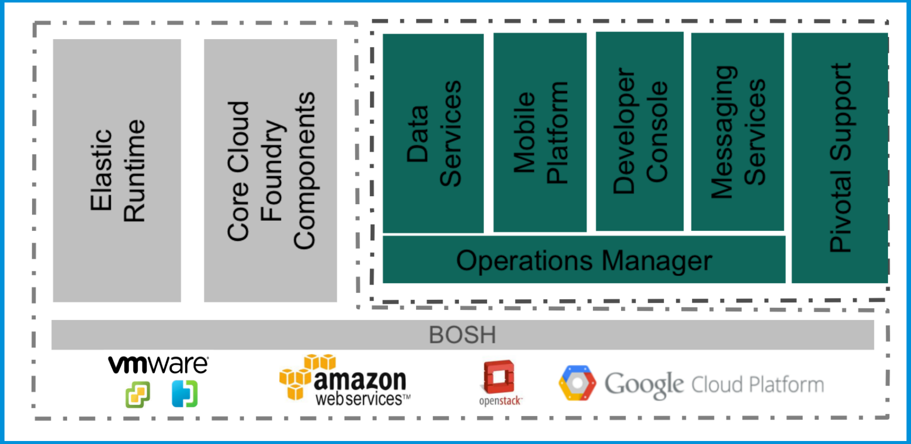
untitled-e6855 .heroku.com

Public — Anyone on the internet can view this app
[✓ Private] — Only collaborators can view this app

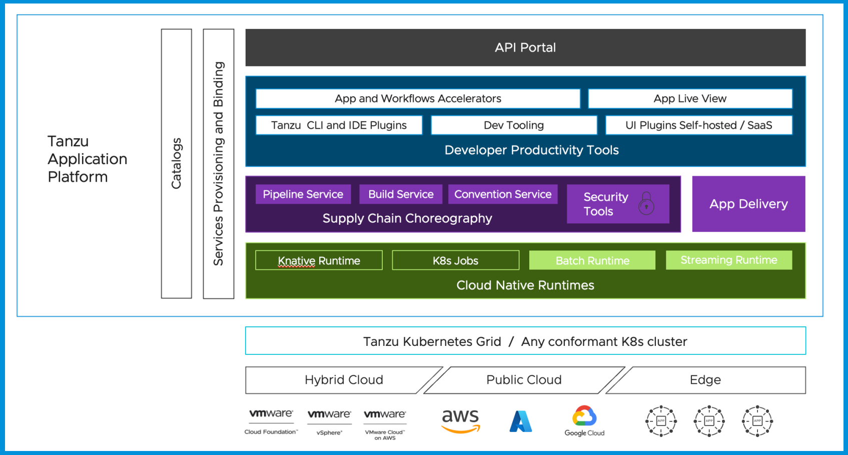
simonstl@simonstl.com Full edit access

Full edit access

2007

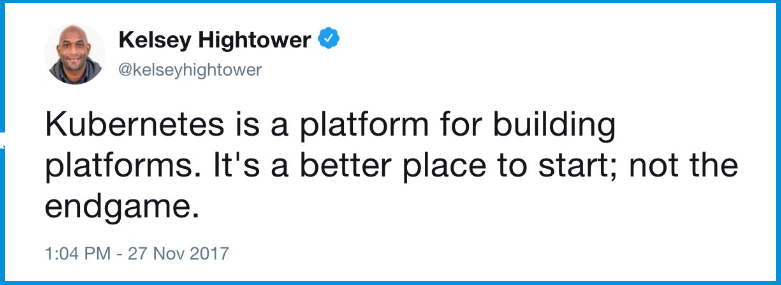


2015



Tanzu Application Platform
Catalogs
Services Provisioning and Binding
API Portal
App and Workflows Accelerators
App Live View
Tanzu CLI and IDE Plugins
Dev Tooling
UI Plugins Self-hosted / SaaS
Developer Productivity Tools
Pipeline Service
Build Service
Convention Service
Security Tools
Supply Chain Choreography
App Delivery
Knative Runtime
K8s Jobs
Batch Runtime
Streaming Runtime
Cloud Native Runtimes
Tanzu Kubernetes Grid / Any conformant K8s cluster
Hybrid Cloud
Public Cloud
Edge
vmware Cloud Foundation
vmware vSphere
vmware VMware Cloud on AWS
aws
Google Cloud

2023 & Beyond



Kelsey Hightower @kelseyhightower

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

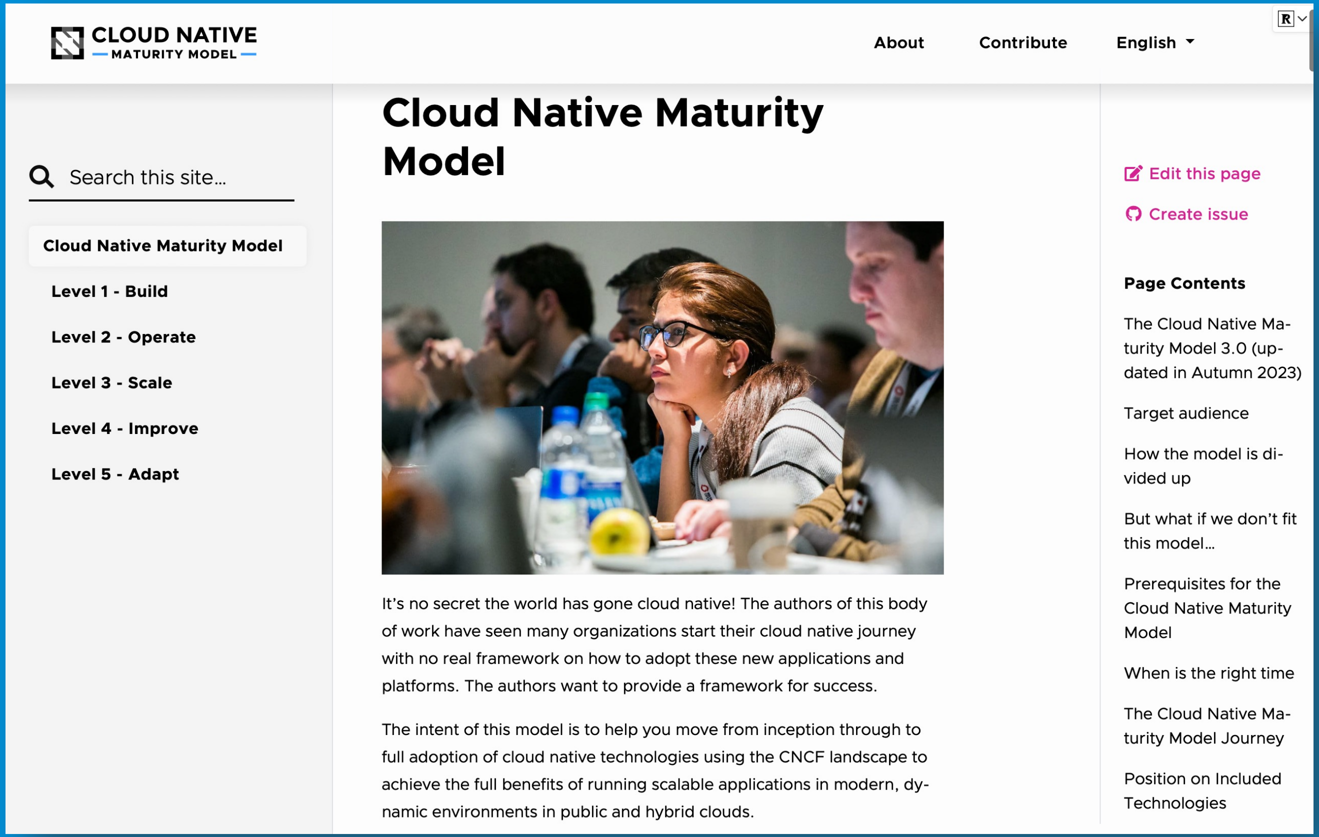
1:04 PM - 27 Nov 2017

Not pictured:

OO, Small Talk, RUP, CORBA, J2EE/.Net, SOA & WS-*, RAD, Low Code, Public Clouds

🍔

The technology is new.
The process is proven.


CLOUD NATIVE MATURITY MODEL

About | Contribute | English

Search this site…

- Cloud Native Maturity Model
  - Level 1 - Build
  - Level 2 - Operate
  - Level 3 - Scale
  - Level 4 - Improve
  - Level 5 - Adapt

Cloud Native Maturity Model

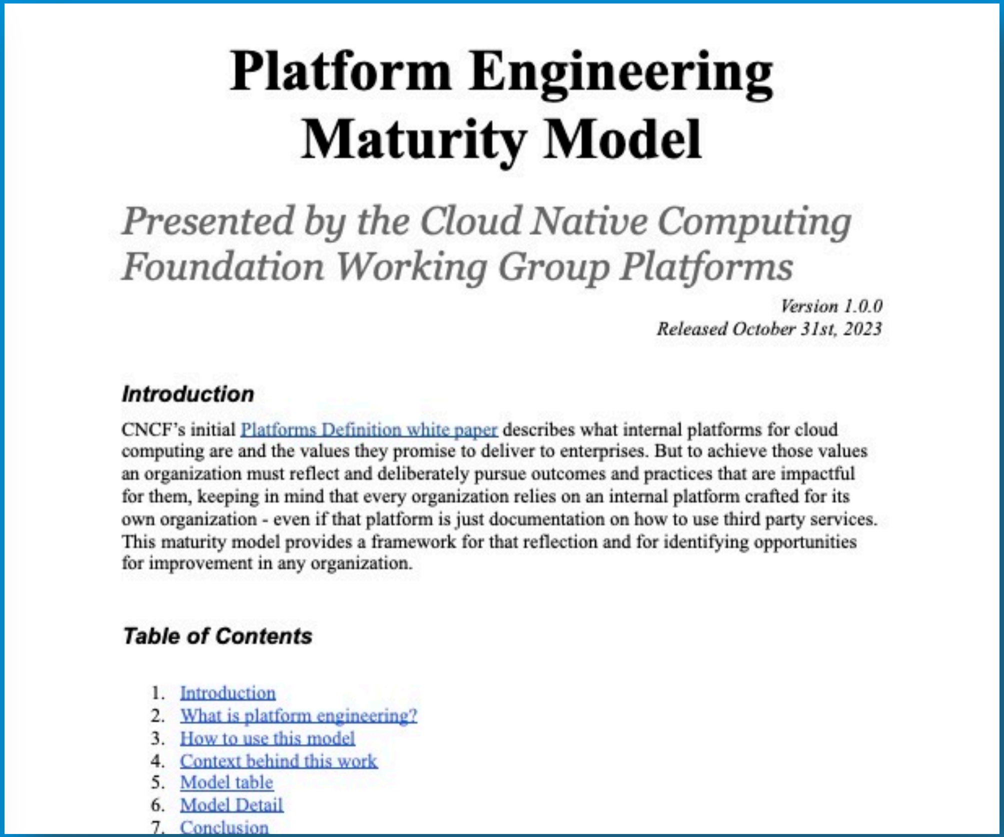
It's no secret the world has gone cloud native! The authors of this body of work have seen many organizations start their cloud native journey with no real framework on how to adopt these new applications and platforms. The authors want to provide a framework for success.

The intent of this model is to help you move from inception through to full adoption of cloud native technologies using the CNCF landscape to achieve the full benefits of running scalable applications in modern, dynamic environments in public and hybrid clouds.

Edit this page

Create issue

Page Contents


Platform Engineering Maturity Model

Presented by the Cloud Native Computing Foundation Working Group Platforms

Version 1.0.0
Released October 31st, 2023

Introduction

CNCF's initial Platforms Definition white paper describes what internal platforms for cloud computing are and the values they promise to deliver to enterprises. But to achieve those values an organization must reflect and deliberately pursue outcomes and practices that are impactful for them, keeping in mind that every organization relies on an internal platform crafted for its own organization - even if that platform is just documentation on how to use third party services. This maturity model provides a framework for that reflection and for identifying opportunities for improvement in any organization.

Table of Contents

Source: "Cloud Native Maturity Model 3.0," CNCF working group, Fall 2023; "Cloud Native Maturity Model Reference Model," CNCF working group, May 2022 to Nov 2023; "The Cloud Native Maturity Model 3.0" Danielle Cook & Simon Forster, KubeCon NA, Nov 2023.

Building a Platform: Cloud Native Maturity Model

Focus	Level 1	Level 2–4	Level 5
Biz	Initial alignment on cloud-native costs and exploration of technology-business relationships.	Progressively integrates cloud-native initiatives with business goals, focusing on ROI, scaling for demand, and optimizing costs while maintaining legacy systems.	Achieves predictable IT spend, with business and technology teams working in harmony towards flexible and adaptable goals.
People	Teams are new to cloud-native, exploring tooling primarily Kubernetes, with basic technical understanding and security considerations.	Expands training in Kubernetes and cloud-native patterns, establishing DevOps practices, and integrating security deeply into the process.	Organization reaches maturity with skilled DevOps and DevSecOps, fostering an environment of experimentation and continuous improvement.
Process	Starts with manual deployments, focusing on application requirements and basic automation.	Develops structured build and deployment processes, embracing CI/CD, and creating feedback loops for continuous improvement and standardization.	Achieves process maturity with cloud-native design capabilities, automated responses to monitoring failures, and optimization of resource usage for cost efficiency.
Policy	Establishes a limited set of documented policies supporting cloud services.	Moves towards standardization and documentation for production, implementing policy-as-code, and defining SLAs around policies and remediation.	Refines policies using advanced technologies, engaging with regulators and the community for continuous improvement in security and compliance.
Tech	Begins with experimentation on Kubernetes and basic cloud-native tooling, assessing fit within the new landscape.	Builds a foundation for production, incorporates monitoring, observability, and security tooling, and scales operations with standardized tools.	Focuses on automation in functional areas, employs operators for operational tasks, and optimizes cloud-native infrastructure and applications for peak efficiency.

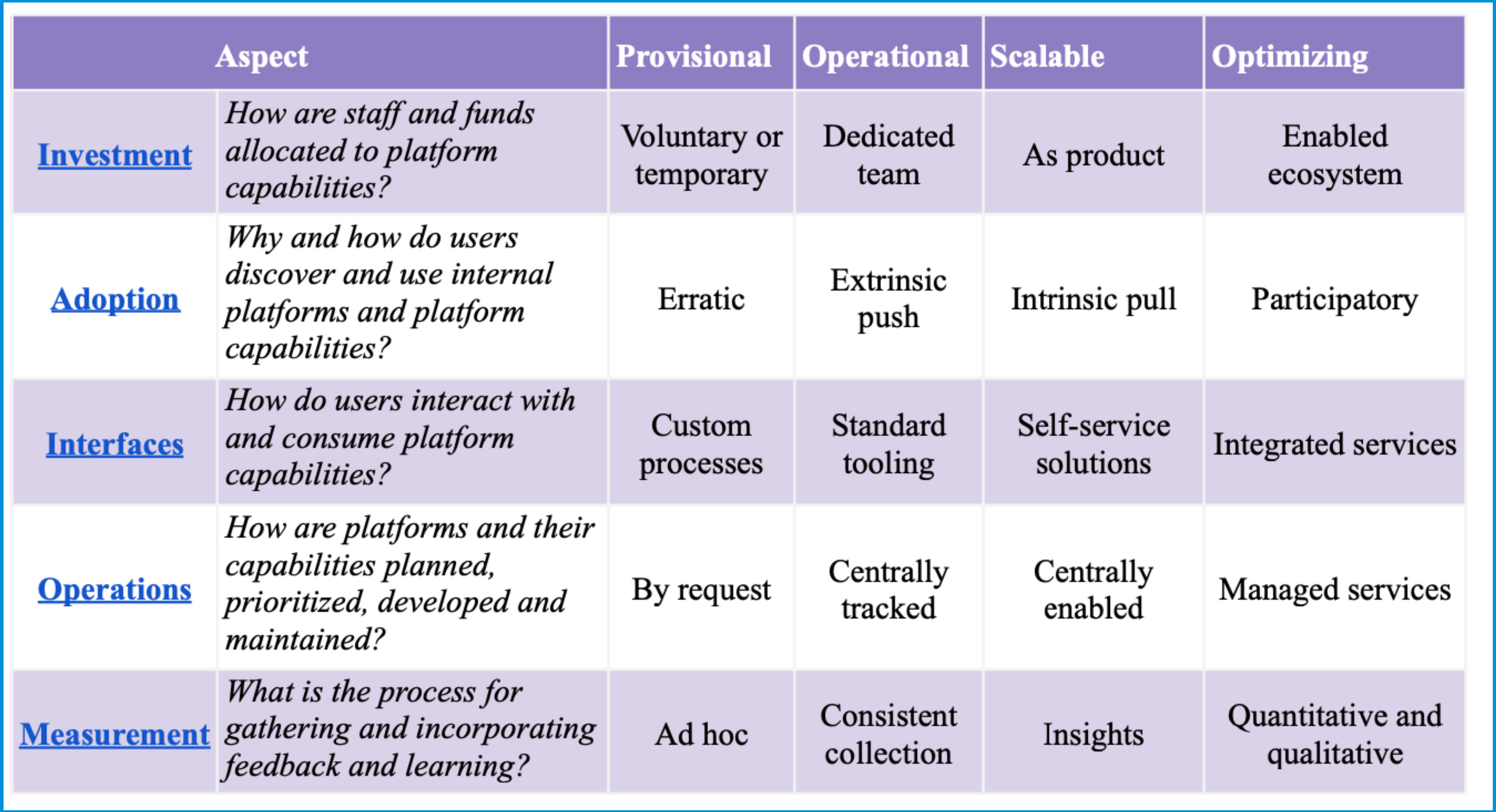
Source: summarized by ChatGPT 4; "Cloud Native Maturity Model 3.0," CNCF working group, Fall 2023; "Cloud Native Maturity Model Reference Model," CNCF working group, May 2022 to Nov 2023; "The Cloud Native Maturity Model 3.0" Danielle Cook & Simon Forster, KubeCon NA, Nov 2023.

Microservices, ops & security as code, dev self-service, dev == biz

	Level 1	Level 2	Level 3	Level 4	Level 5
People	Exploring w/ MVPs; learning cloud native concepts; learning **12 factor & microservices**.	New teams, learning loops, test **centralization, standardization**, CI/CD, devs k8s skills.	Skills better; silos to balanced teams; intro DevSecOps into dev org.	Cloud is default, **dev self-service, feedback loop from app to biz**, security shift-left.	All cloud native, **full self-service, biz ownership of apps**, resilience, security-minded, FinOps approach.
Technology	**Pre-prod**; building cloud platform w/IaC; **learning k8s with PoC**; launch DevOps.	**Apps in prod**; k8s for prod w/security & reliability; microservices apps required; **CI & tools for prod deployment**; k8s services for secrets, encryption, AA.	Focus on "observability"; **collect cost metrics**; early GitOps for prod; servers->services; standardize ops tools; **automate security policy & enforcement**.	k8s expert; microservices/API app design; full k8s management, alerting; **fix prod with CI/CD**.	Full IaC; **all new apps cloud native**; modernizing legacy apps; **prod auto-remediation**; full GitOps; ML security.

Source: summarized by ChatGPT 4; "Cloud Native Maturity Model 3.0," CNCF working group, Fall 2023; "Cloud Native Maturity Model Reference Model," CNCF working group, May 2022 to Nov 2023; "The Cloud Native Maturity Model 3.0" Danielle Cook & Simon Forster, KubeCon NA, Nov 2023.

Running a platform: Platform Engineering Maturity Model



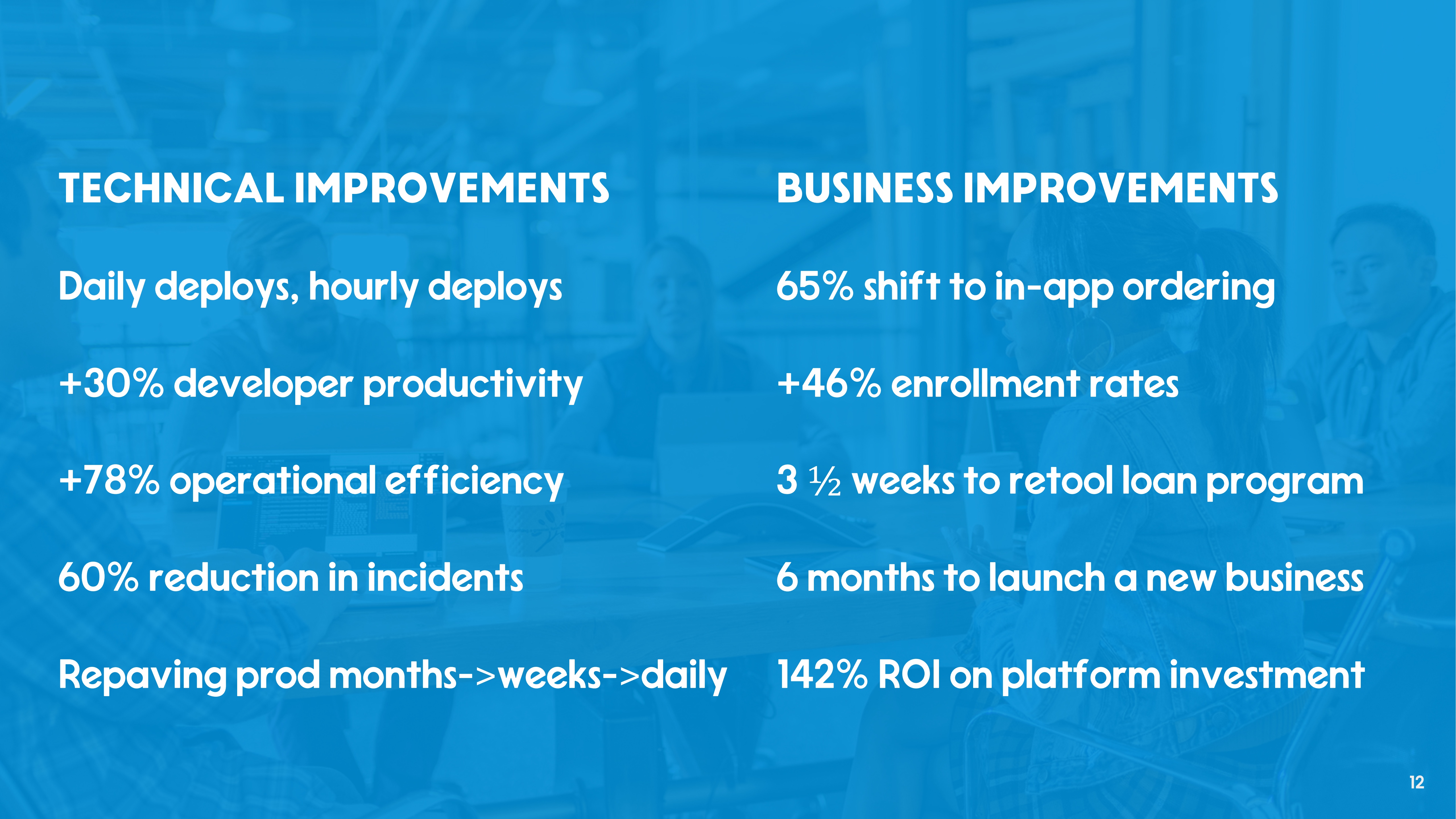
Aspect		Provisional	Operational	Scalable	Optimizing
Investment	*How are staff and funds allocated to platform capabilities?*	Voluntary or temporary	Dedicated team	As product	Enabled ecosystem
Adoption	*Why and how do users discover and use internal platforms and platform capabilities?*	Erratic	Extrinsic push	Intrinsic pull	Participatory
Interfaces	*How do users interact with and consume platform capabilities?*	Custom processes	Standard tooling	Self-service solutions	Integrated services
Operations	*How are platforms and their capabilities planned, prioritized, developed and maintained?*	By request	Centrally tracked	Centrally enabled	Managed services
Measurement	*What is the process for gathering and incorporating feedback and learning?*	Ad hoc	Consistent collection	Insights	Quantitative and qualitative

Source: "Platform Engineering Maturity Model," CNCF Platforms Working Group, October, 2023.

Getting ready for a long journey

Questions, topics, & analysis

1. Expect 3 to 5 years: “Small organizations might swim through level three, while large enterprise organizations with lots of heritage and legacy, might be here for years.”
2. Do you have and/or need microservices and 12 factor apps? How many new apps do you have?
3. How will you modernize your existing apps? Would lift-and-shift or leave alone be better?
4. How will you standardize & centralize infrastructure, [dev|ops|security|compliance] practices?
5. How can you make continuous learning stick?
6. How will you change the organization and team definition, structures, and incentives?
7. How will you get The Business to work weekly with the developers?
8. Costs do not decrease until you decommission old infrastructure and legacy apps.



TECHNICAL IMPROVEMENTS

Daily deploys, hourly deploys

+30% developer productivity

+78% operational efficiency

60% reduction in incidents

Repaving prod months->weeks->daily

BUSINESS IMPROVEMENTS

65% shift to in-app ordering

+46% enrollment rates

3 ½ weeks to retool loan program

6 months to launch a new business

142% ROI on platform investment

🧰

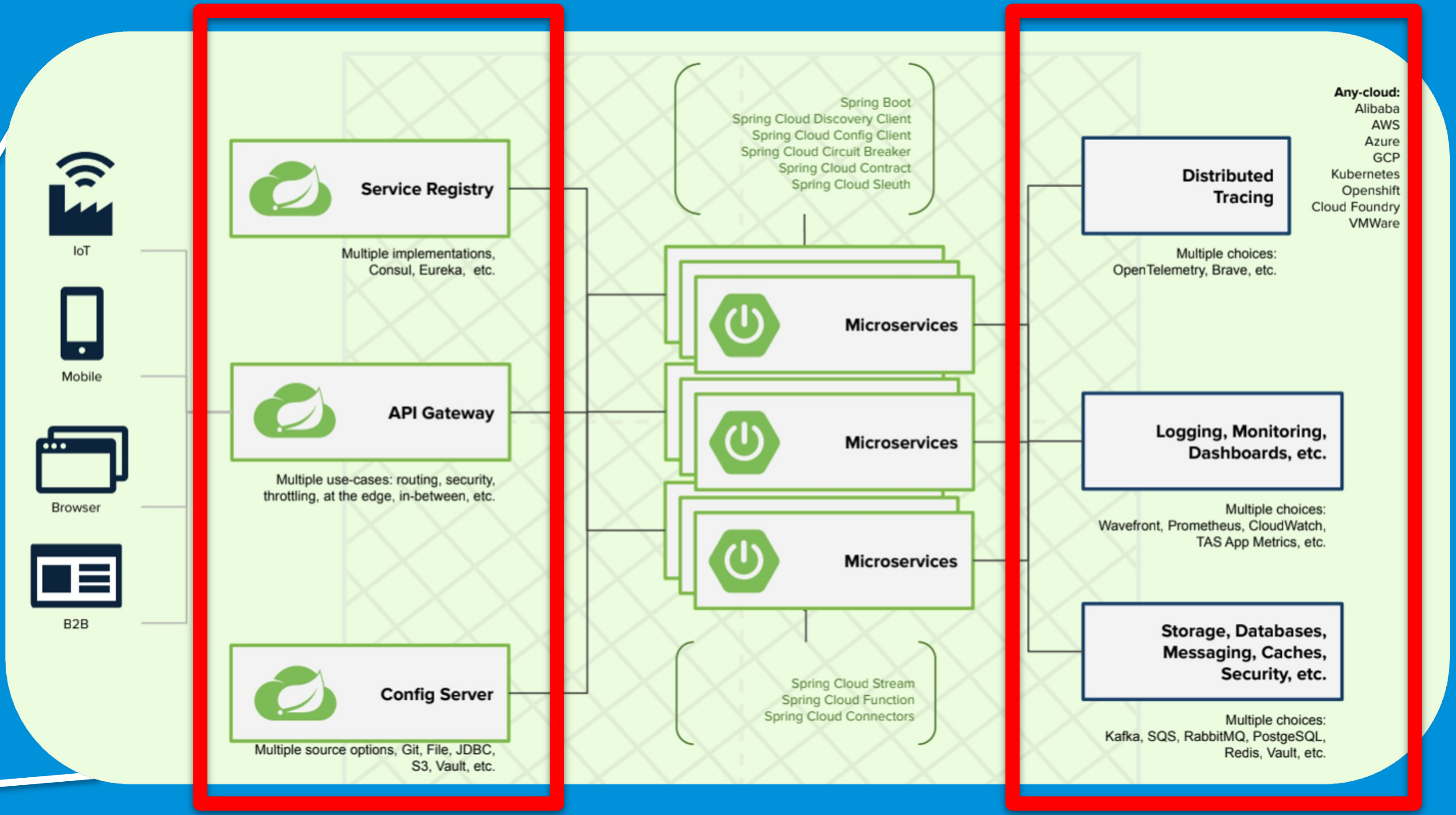
Lessons learned from 8 years of platform engineering, a selection

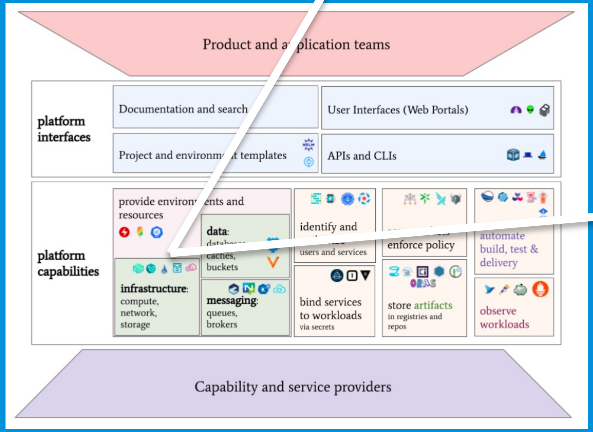
Build it and they don’t come

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

Thomas Müller, Mercedes-Benz

The platform team should manage undifferentiated services


IoT
Mobile
Browser
B2B
Service Registry
Multiple implementations, Consul, Eureka, etc.
API Gateway
Multiple use-cases: routing, security, throttling, at the edge, in-between, etc.
Config Server
Multiple source options, Git, File, JDBC, S3, Vault, etc.
Spring Boot
Spring Cloud Discovery Client
Spring Cloud Config Client
Spring Cloud Circuit Breaker
Spring Cloud Contract
Spring Cloud Sleuth
Microservices
Microservices
Microservices
Spring Cloud Stream
Spring Cloud Function
Spring Cloud Connectors
Distributed Tracing
Multiple choices: OpenTelemetry, Brave, etc.
Logging, Monitoring, Dashboards, etc.
Multiple choices: Wavefront, Prometheus, CloudWatch, TAS App Metrics, etc.
Storage, Databases, Messaging, Caches, Security, etc.
Multiple choices: Kafka, SQS, RabbitMQ, PostgeSQL, Redis, Vault, etc.
Any-cloud:
Alibaba
AWS
Azure
GCP
Kubernetes
Openshift
Cloud Foundry
VMWare


Product and a plication teams
platform interfaces
Documentation and search
User Interfaces (Web Portals)
Project and environm t templates
APIs and CLIs
platform capabilities
provide environ ents and resources
data: databases, caches, buckets
infrastructure: compute, network, storage
messaging: queues, brokers
identify and users and services
bind services to workloads via secrets
enforce policy
store artifacts in registries and repos
automate build, test & delivery
observe workloads
Capability and service providers

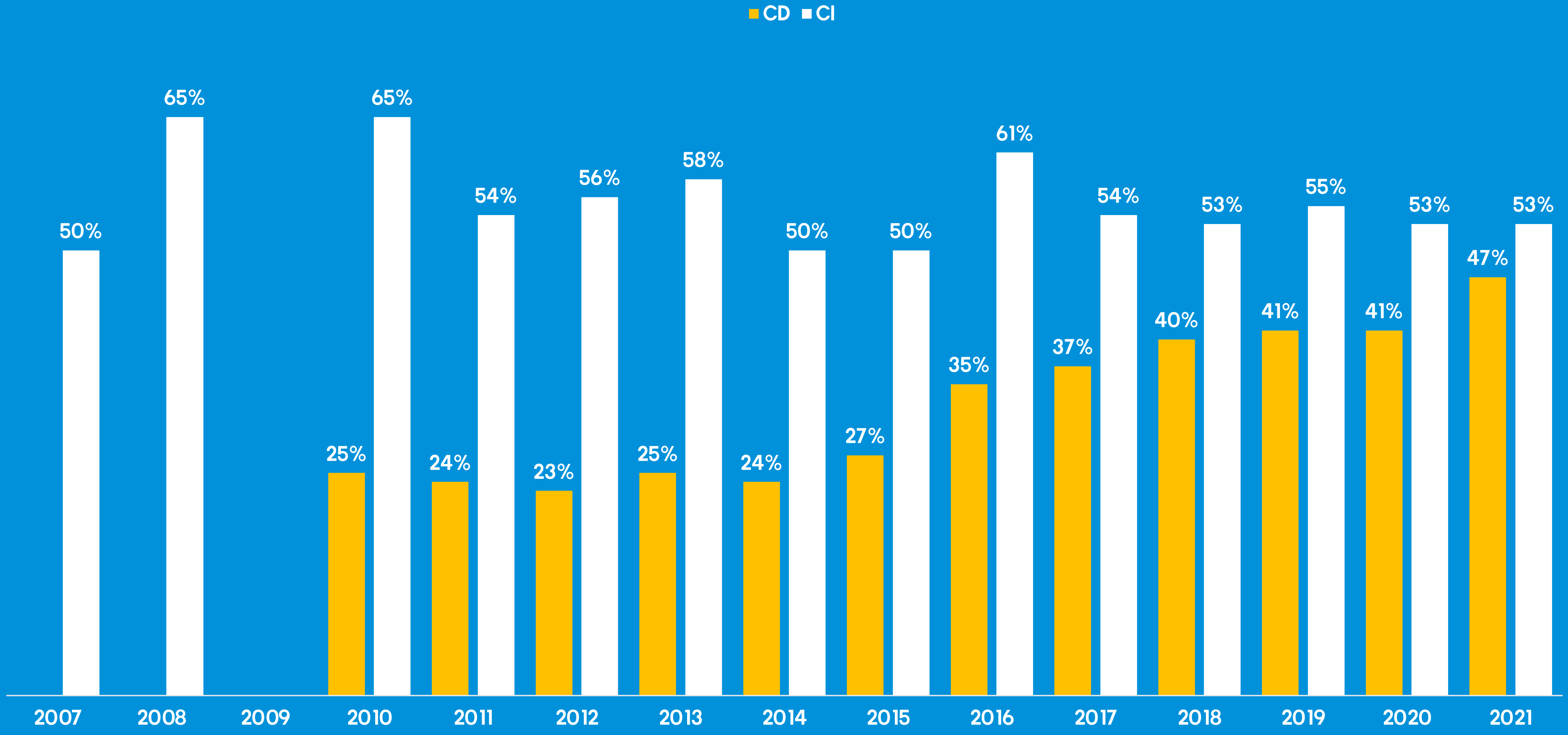
Find the Developer Toil, Confusion, Blockers

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.

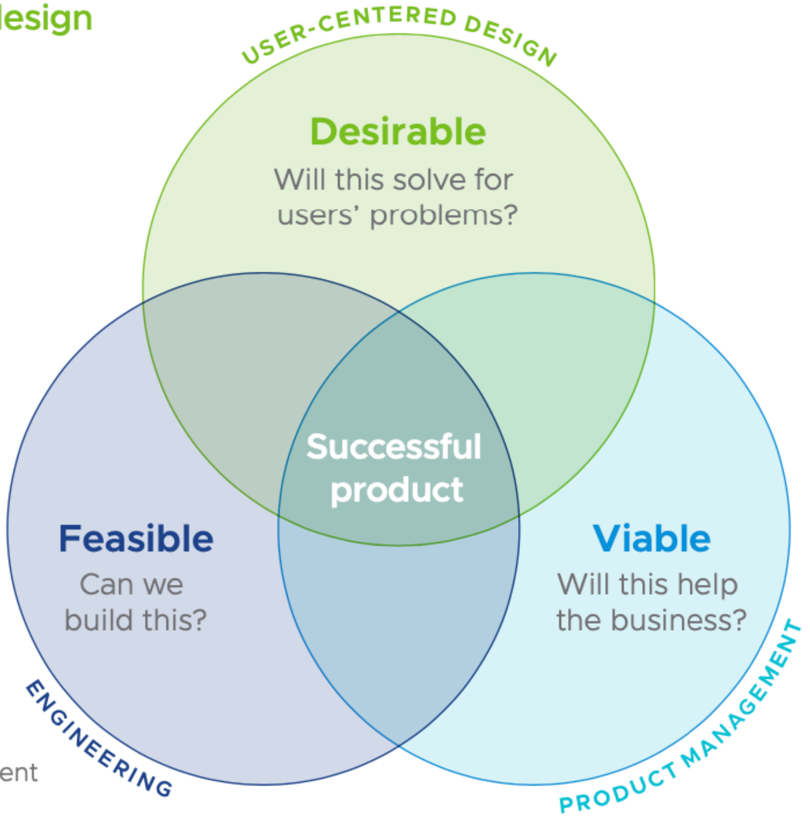
Sources: "Developer Toil: The Hidden Tech Debt," Susie Forbath, Tyson McNulty, and Coté, August, 2022. See also Michael Galloway's interview questions for platform product managers.

CI and CD usage, 2007 to 2021



Sources: State of Agile Surveys, VersionOne/CollabNet/digital.ai.

Platform team composition mirrors app dev



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 interations
CI / CD

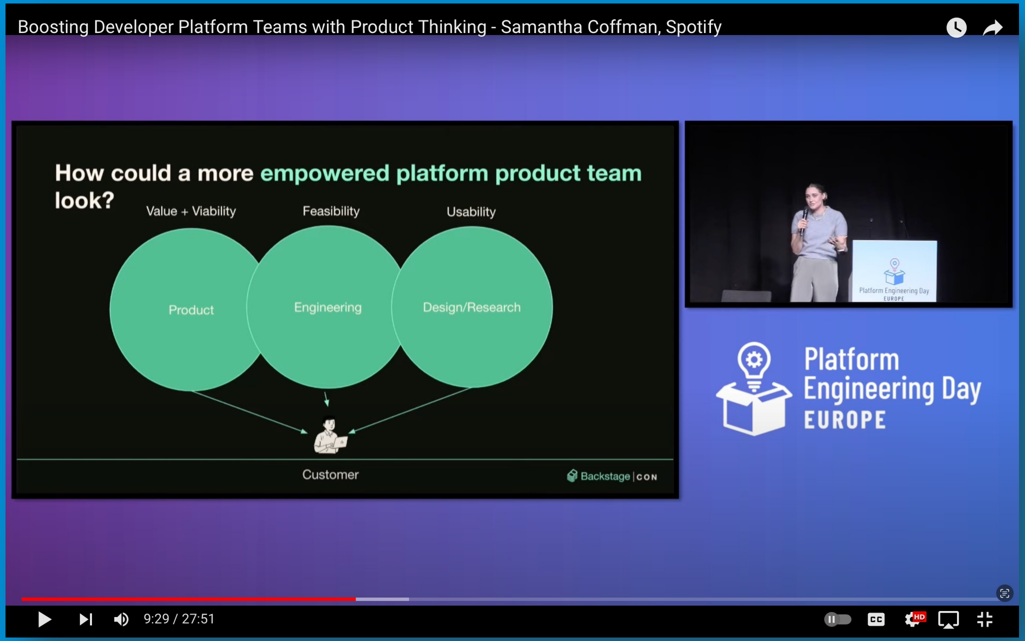
Lean Startup

ACTIVITIES:

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

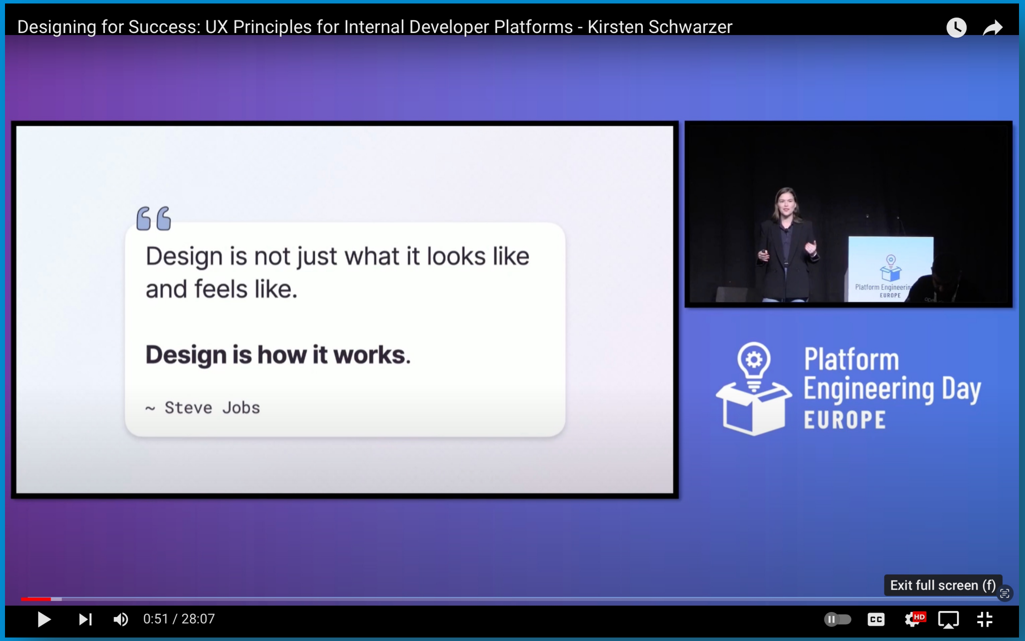
Sources: “The Product Manager Playbook,” VMware Tanzu Labs.

Design is best practice outside of Tanzu too



Boosting Developer Platform Teams with Product Thinking - Samantha Coffman, Spotify

How could a more **empowered platform product team** look?



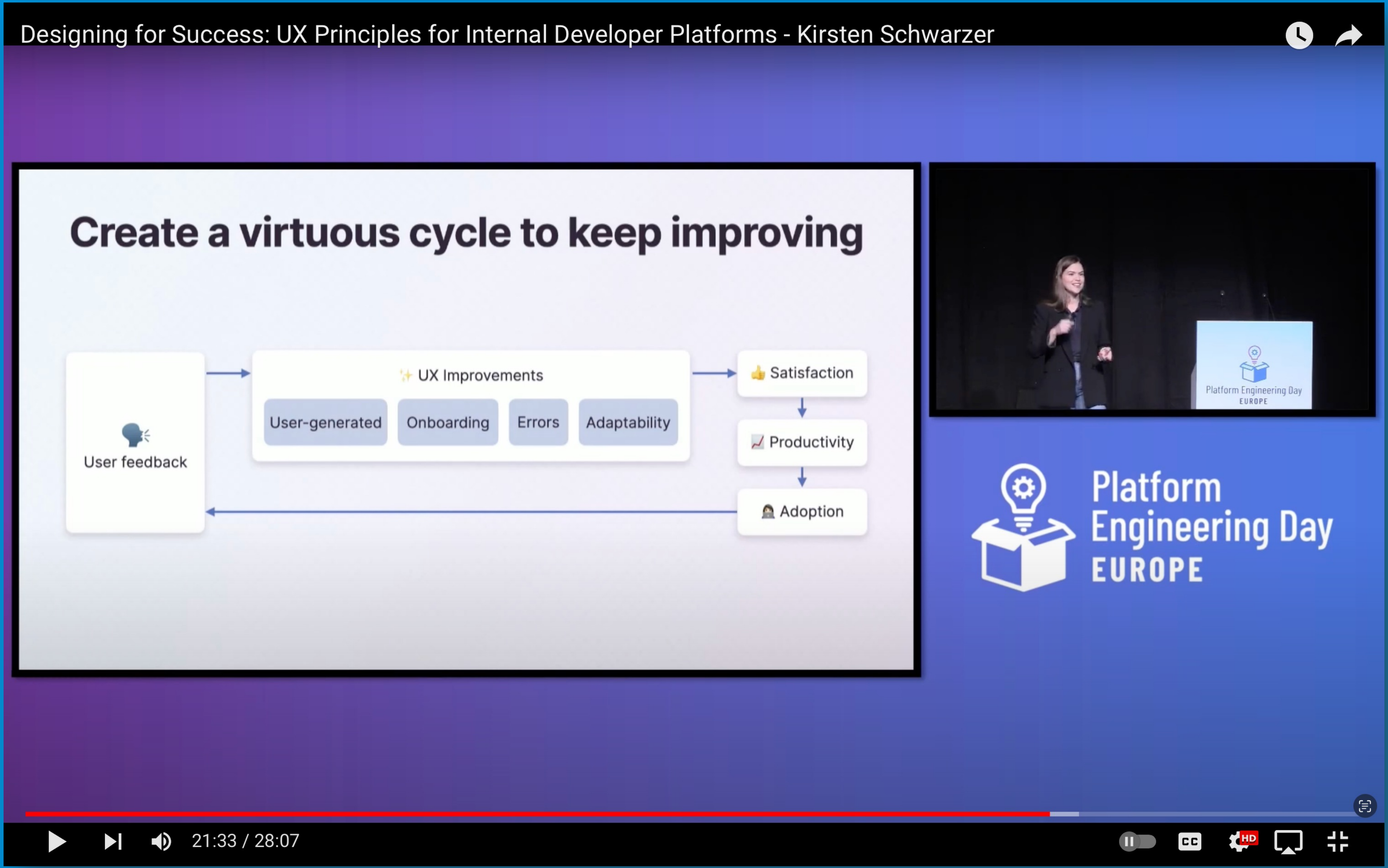
Designing for Success: UX Principles for Internal Developer Platforms - Kirsten Schwarzer

Design is not just what it looks like and feels like.

Design is how it works.

~ Steve Jobs

Source: "Boosting Developer Platform Teams with Product Thinking," Samantha Coffman, KubeCon EU, March 2024; "Designing for Success: UX Principles for Internal Developer Platforms," Kirsten Schwarzer, KubeCon EU, March, 2024.



Designing for Success: UX Principles for Internal Developer Platforms - Kirsten Schwarzer

Create a virtuous cycle to keep improving

Source: "Designing for Success: UX Principles for Internal Developer Platforms," Kirsten Schwarzer, KubeCon EU, March, 2024.

Platform marketing, advocacy, consulting

Organizational Learning

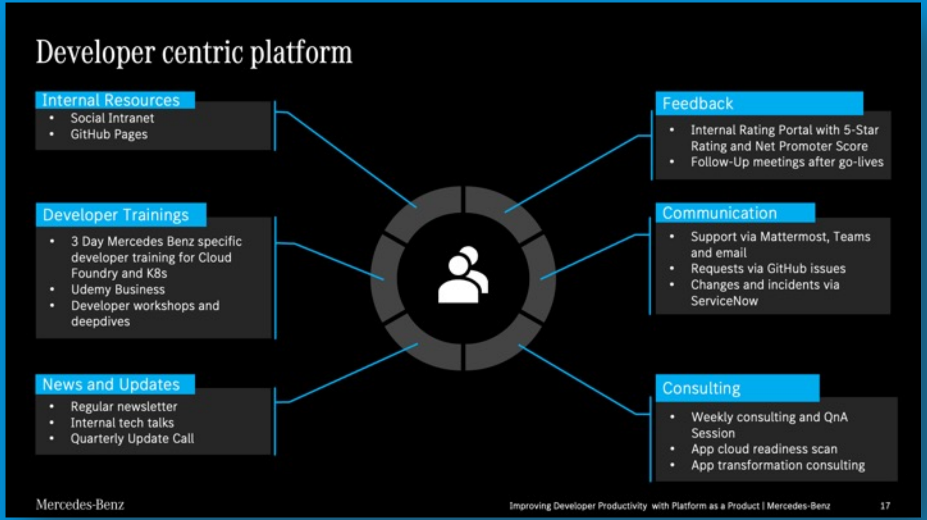
Focus on ways of working…..

DUKE ENERGY.

INNOVATION CENTER
AT OPTIMIST HALL

GUIDE TO
EVERYTHING

VERSION 1.0


Developer centric platform

Internal Resources
- Social Intranet
- GitHub Pages

Developer Trainings
- 3 Day Mercedes Benz specific developer training for Cloud Foundry and K8s
- Udemy Business
- Developer workshops and deepdives

News and Updates
- Regular newsletter
- Internal tech talks
- Quarterly Update Call

Feedback
- Internal Rating Portal with 5-Star Rating and Net Promoter Score
- Follow-Up meetings after go-lives

Communication
- Support via Mattermost, Teams and email
- Requests via GitHub issues
- Changes and incidents via ServiceNow

Consulting
- Weekly consulting and QnA Session
- App cloud readiness scan
- App transformation consulting

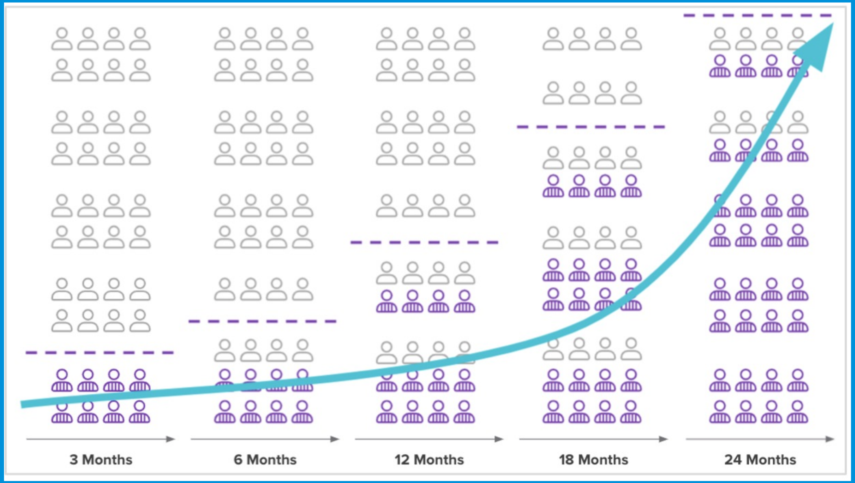
Mercedes-Benz   Improving Developer Productivity with Platform as a Product | Mercedes-Benz   17

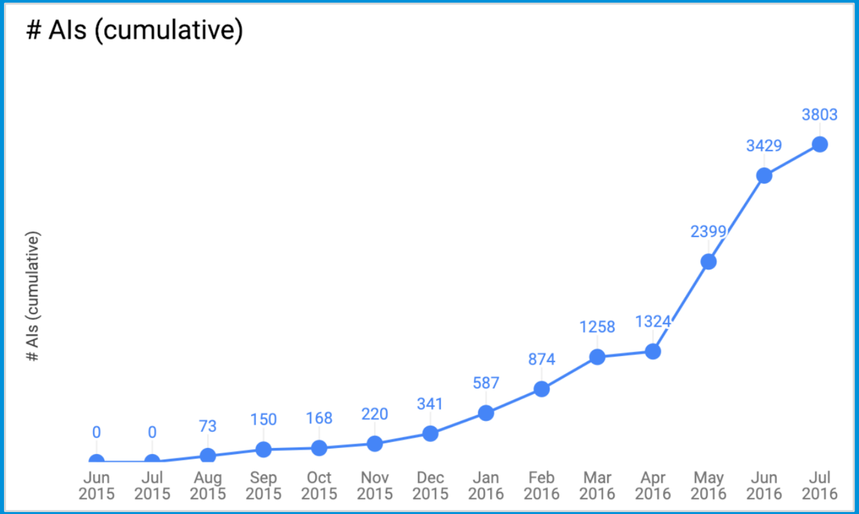
Sources: ING, 2023;BT Canvas team; MB.io; Duke Energy; Allstate; "Take DevOps to 11 and Sprinkle Cloud on it with Rainbows and Unicorns," Matt Curry, s1p 2017. "Improve Developer Productivity with Platform as a Product," VMware Explore, Nov. 2022.

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





HOME DEPOT TIMELINE

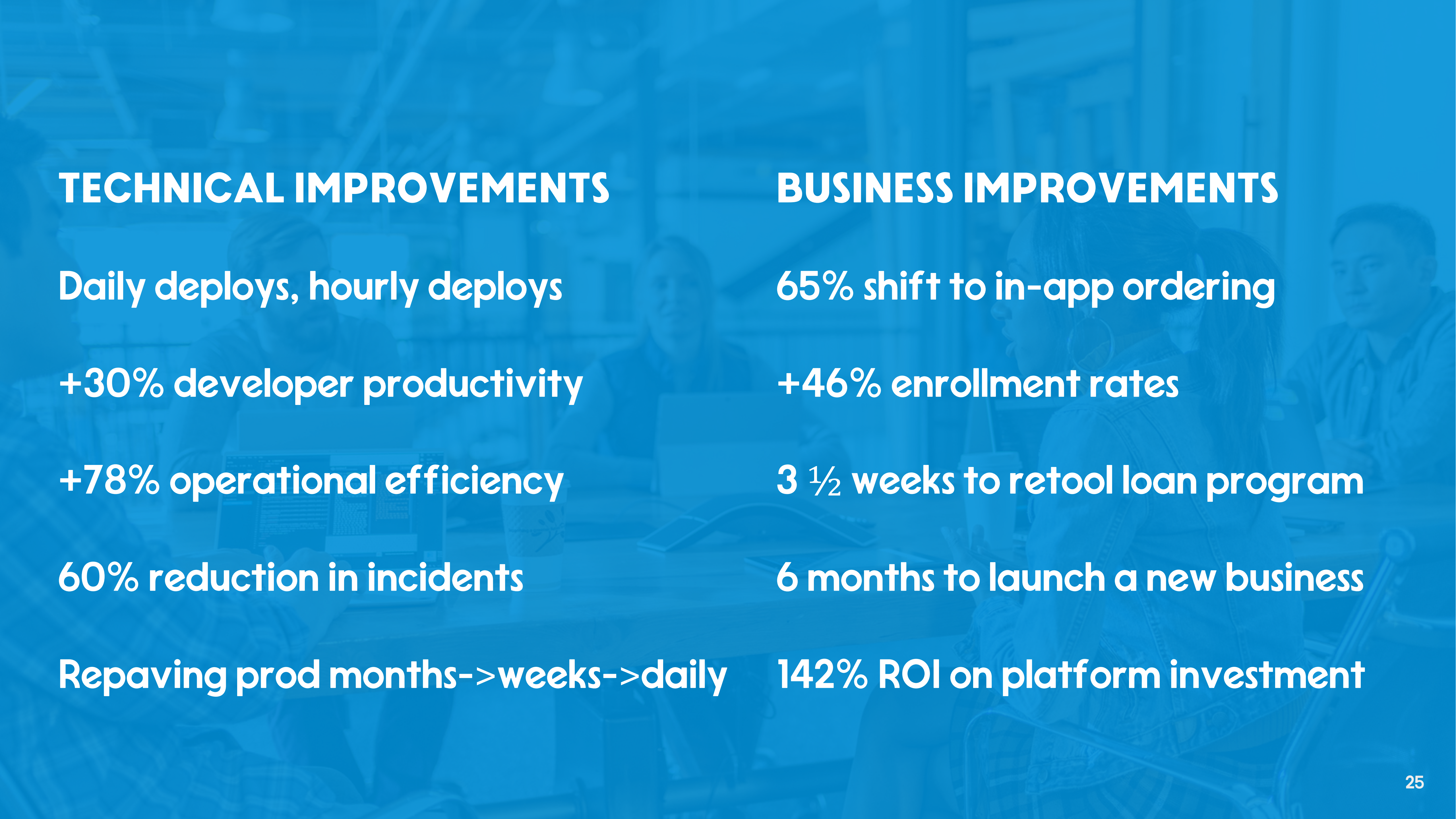
2015: Handful of apps, e.g., paint desk, tool rental

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 Home Depot 2022[?]Q4 earnings call; *The Business Bottleneck*, Coté.



TECHNICAL IMPROVEMENTS

Daily deploys, hourly deploys

+30% developer productivity

+78% operational efficiency

60% reduction in incidents

Repaving prod months->weeks->daily

BUSINESS IMPROVEMENTS

65% shift to in-app ordering

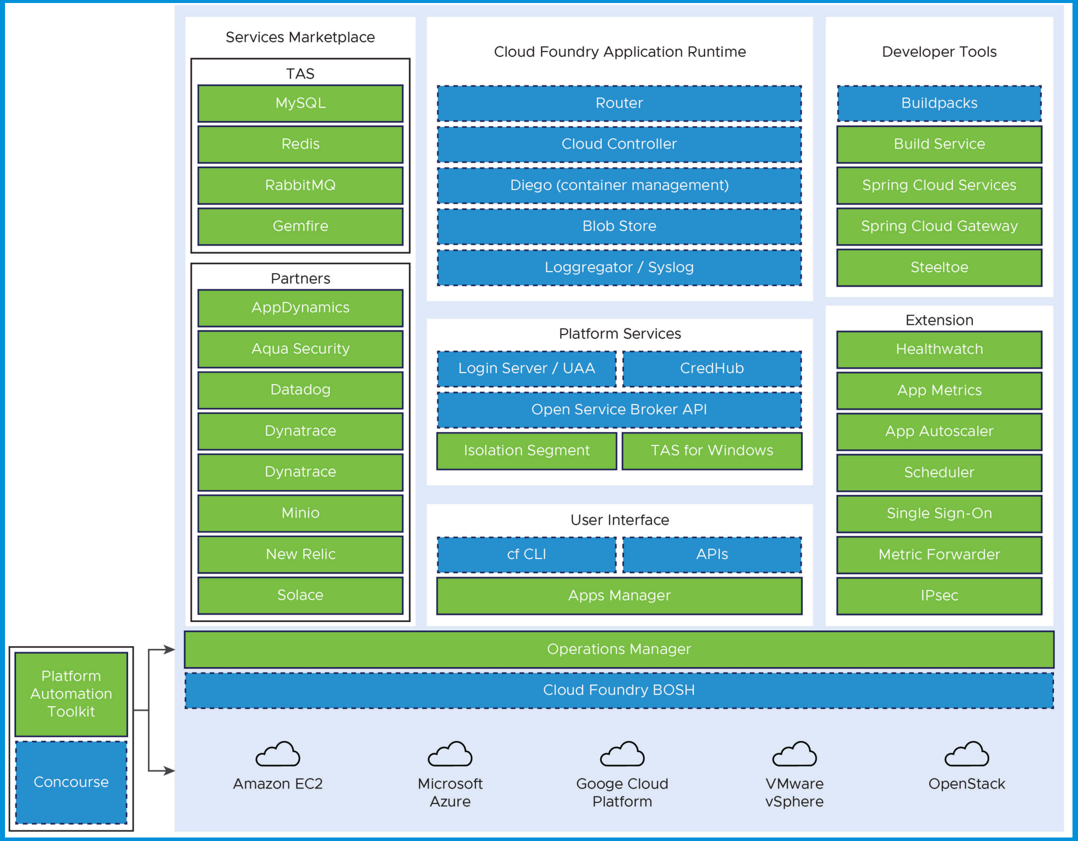
+46% enrollment rates

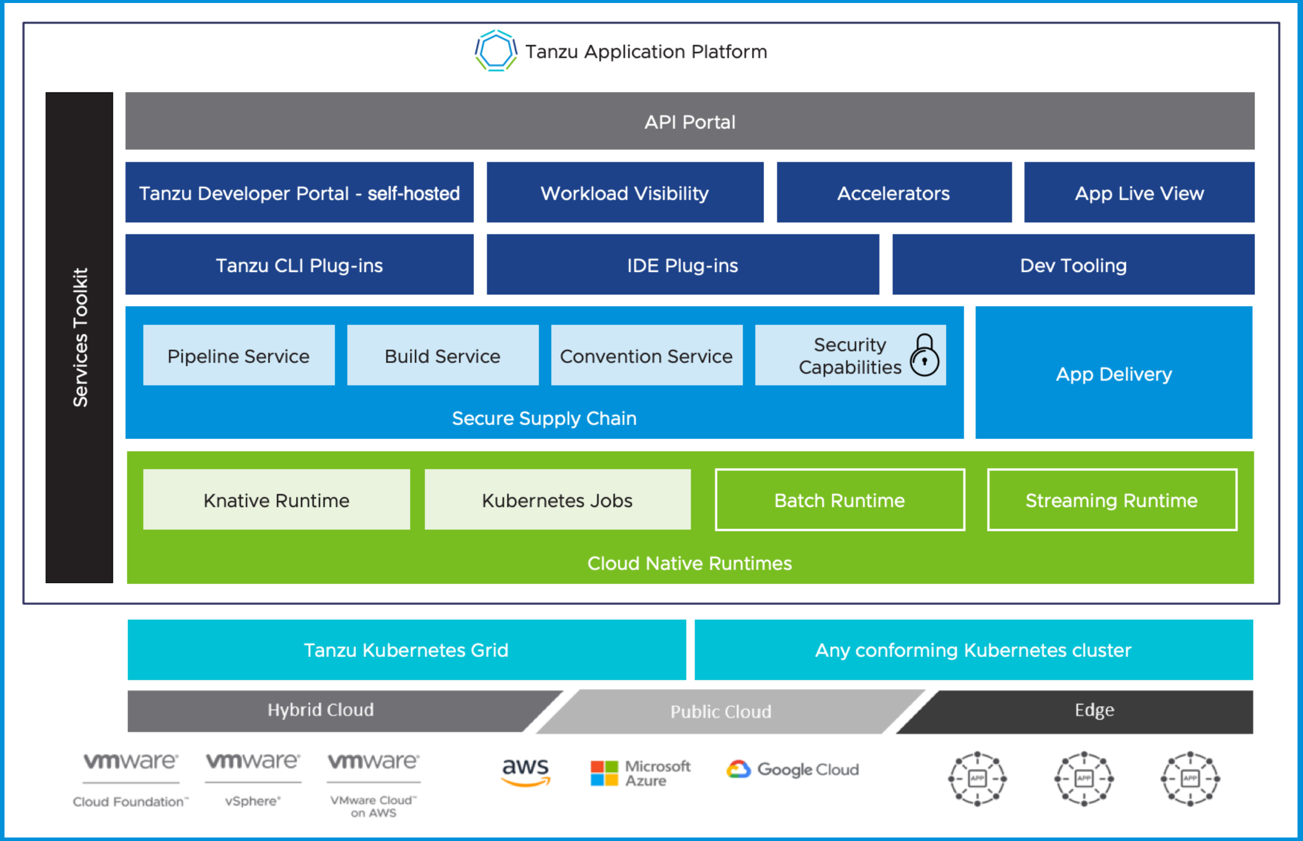
3 ½ weeks to retool loan program

6 months to launch a new business

142% ROI on platform investment

A fine selection of pre-shaved yaks https://tanzu.vmware.com





Thanks!

📥 https://newsletter.cote.io

📚 https://cote.io/platform/

🏢 cote@broadcom.com