Tanzu®
by Broadcom

Modernizing app, platforms, & culture

Maybe it’s not as hard as it seems

Coté

Feb 2024

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

We all know that

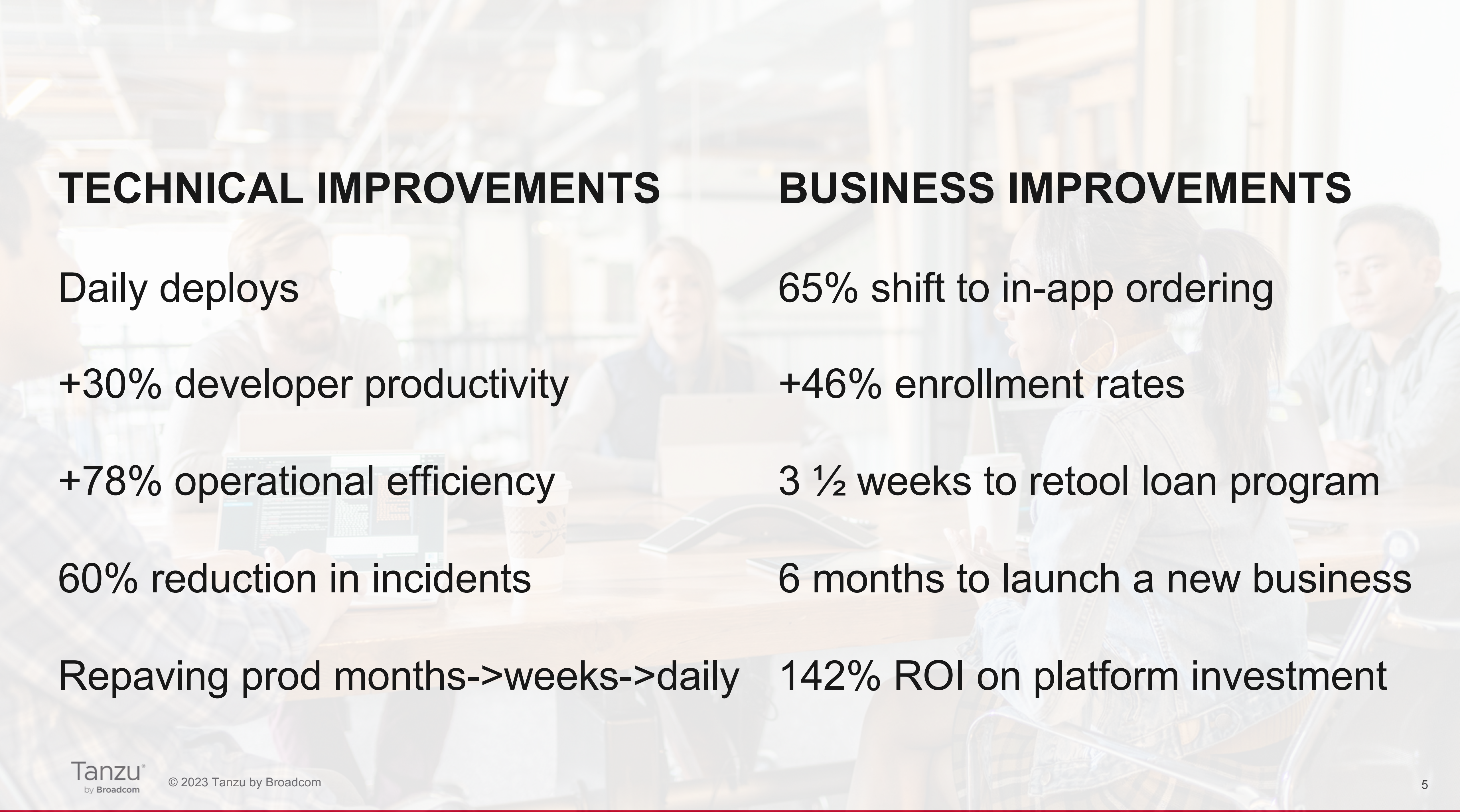
Changing organizations fails 70% of the time.

Sources: "Mind-sets matter in transformations," McKinsey, 2019, many other sources; "How Studying Organizational Change Lost Its Way," Journal of Change Management, Mark Hughes, 2022.

Actually,

We have no idea how frequently organization change fails *or succeeds.*

Sources: "Mind-sets matter in transformations," McKinsey, 2019, many other sources; "How Studying Organizational Change Lost Its Way," Journal of Change Management, Mark Hughes, 2022.



TECHNICAL IMPROVEMENTS

Daily 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

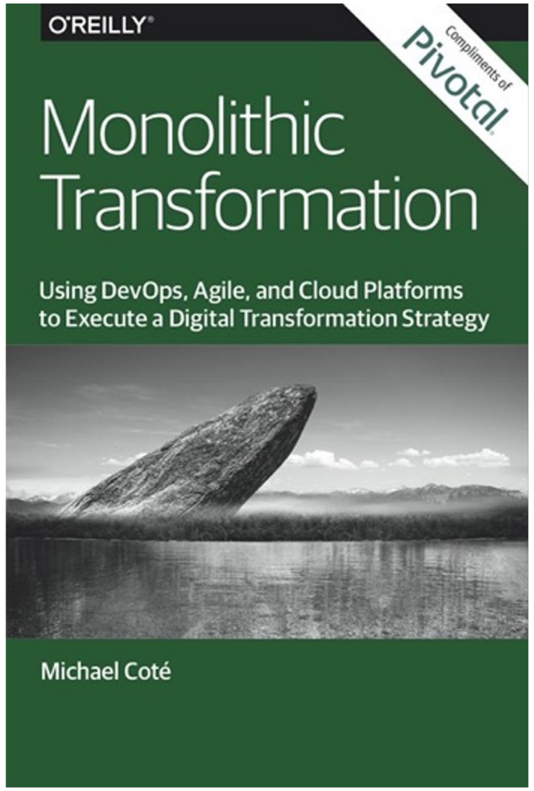
3 ½ weeks to retool loan program

6 months to launch a new business

142% ROI on platform investment

Coté

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

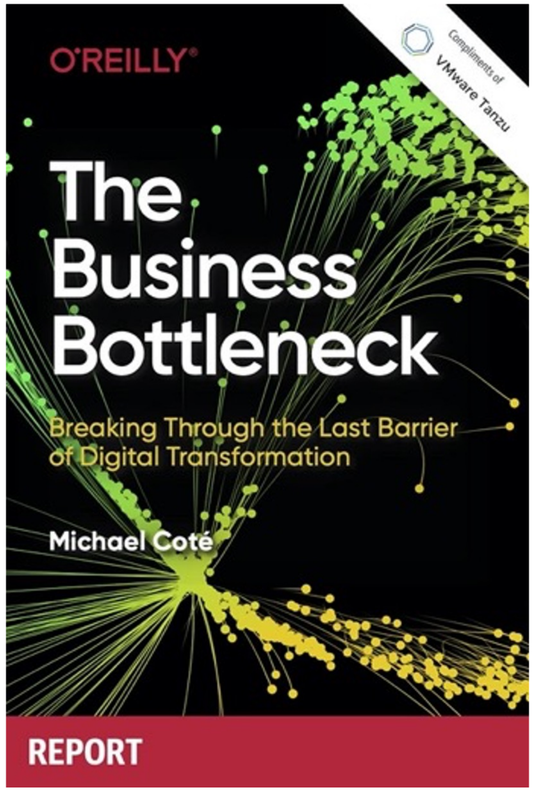

O'REILLY

Compliments of Pivotal

Monolithic Transformation

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

Michael Coté


O'REILLY

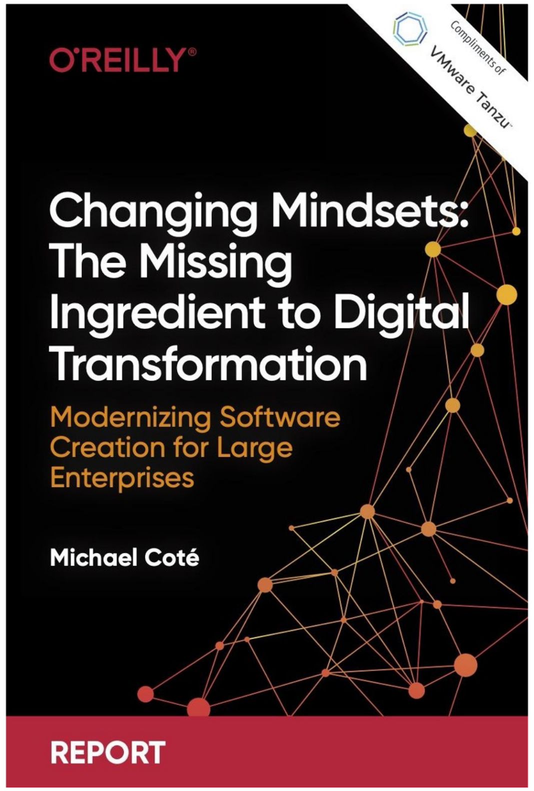
Compliments of VMware Tanzu

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

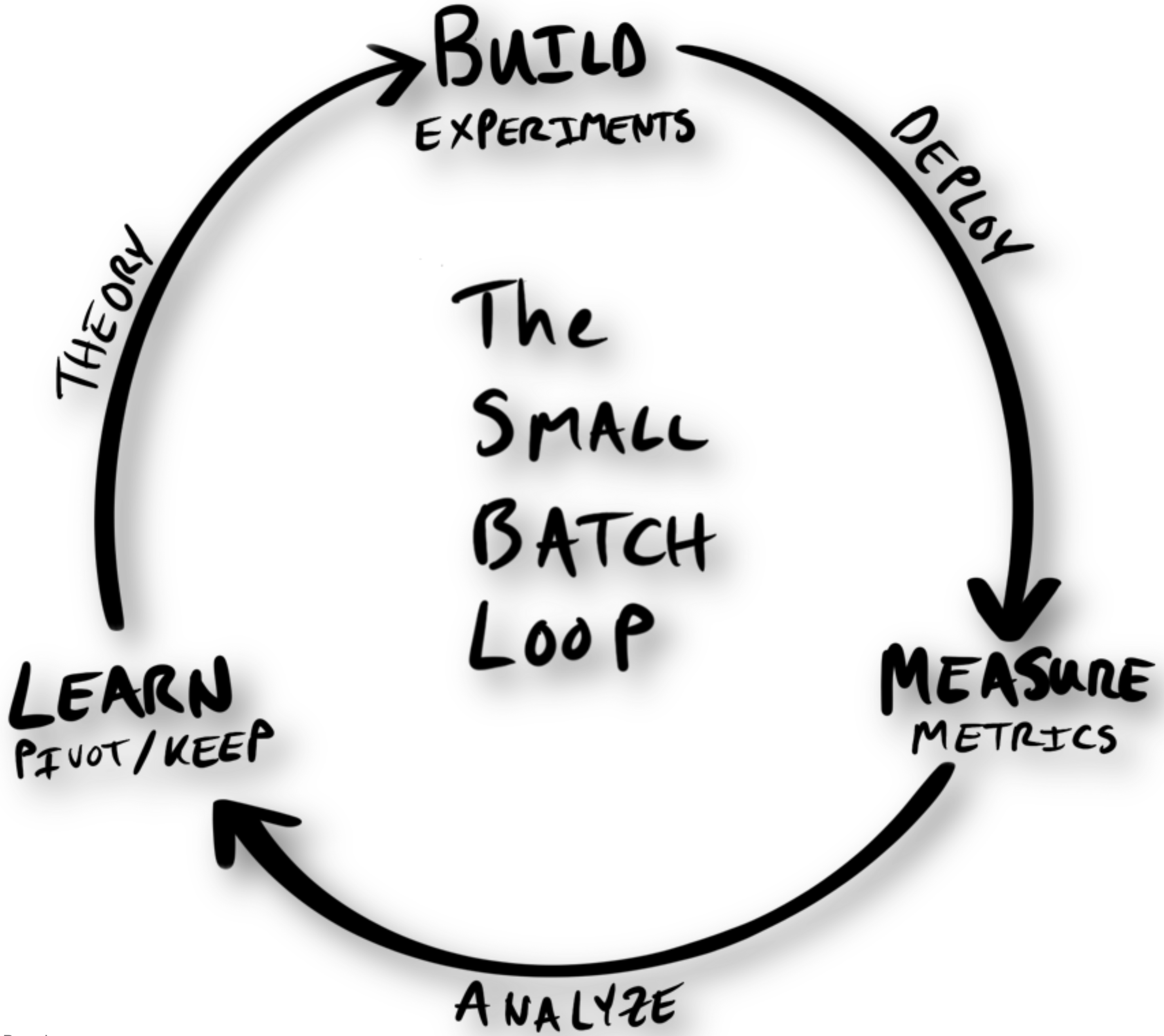
REPORT

Tanzu Talk

SOFTWARE DEFINED TALK

RedMonk DELL 451 Research Pivotal vmware Tanzu by Broadcom

App Dev 👉 Product


BUILD
EXPERIMENTS
DEPLOY
MEASURE
METRICS
ANALYZE
LEARN
PIVOT / KEEP
THEORY
The Small Batch Loop

People are:

- Innovative
- ~~Risk takers~~ Like Learning
- People-centric

Leaders give them:

- Autonomy
- Trust
- Voice

The Product Development Toolbox

User Centered Design

Understand the user and their needs and problems.

PRACTICES

- User Interviews
- Ethnographic studies
- Persona definition
- Prototype creation

Lean Product Management

Avoid building the wrong thing. Easily change direction if needed.

PRACTICES

- MVP definition
- Lean experiments
- Test assumptions
- Data driven decisions

Extreme Programming

Code at a consistent speed and quality in the face of changing requirements.

PRACTICES

- Pair Programming
- TDD
- Short iterations
- CI / CD

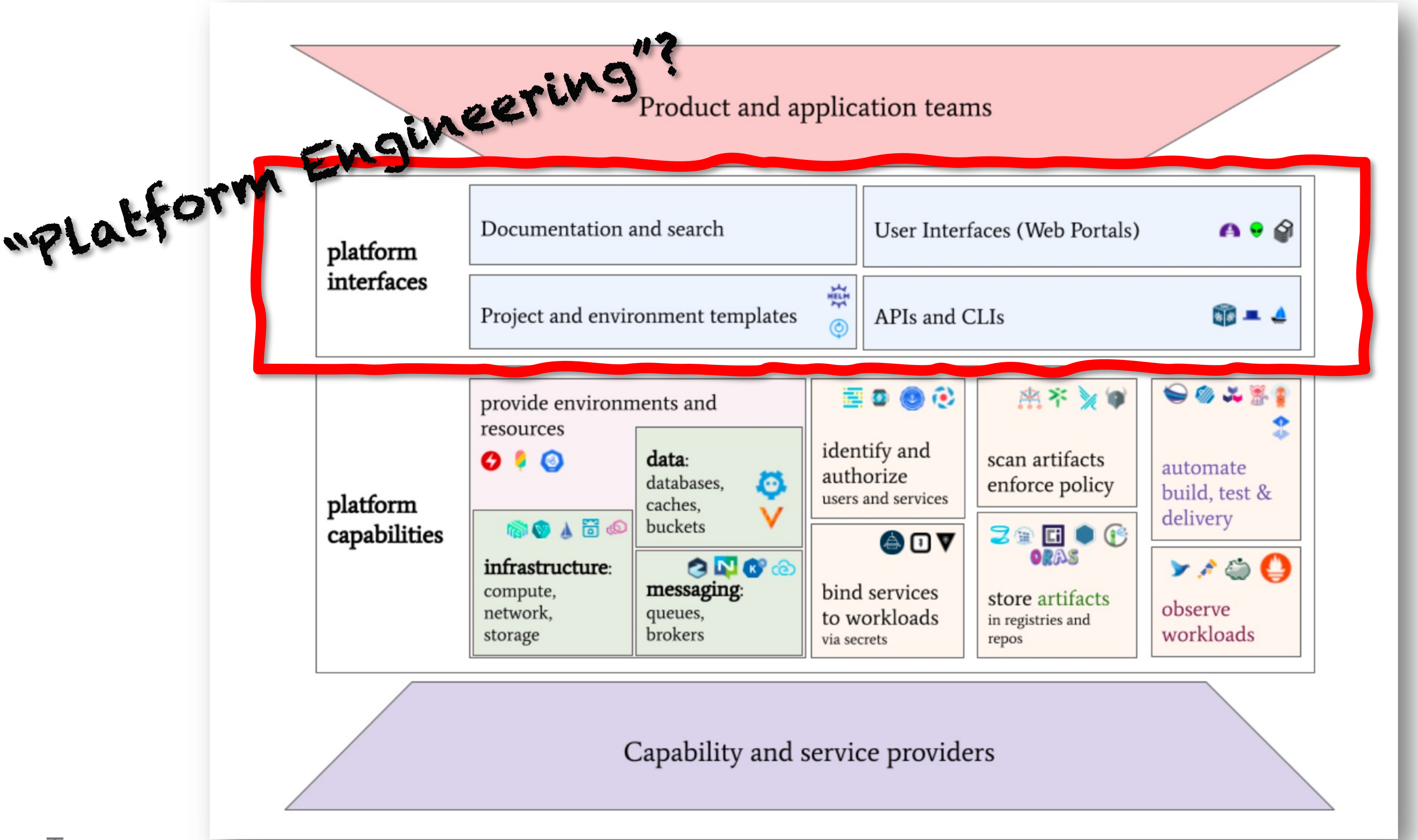
Enterprise Architecture

Understand how the system wants to behave. Architect for constant iteration.

PRACTICES

- Event Storming
- Boris
- SNAP
- Patterns

Infrastructure 👉 Platforms


"Platform Engineering"?
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

Source: “CNCF Platforms White Paper,” March 2023.

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

Thomas Müller, Mercedes-Benz

Find the Developer Toil, Confusion, Blockers

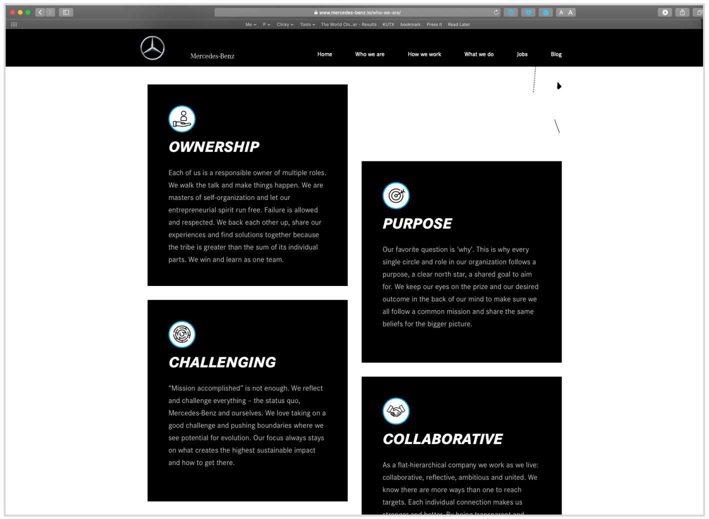
1. What are we making?
2. We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
3. I have the context I need to confidently make changes while I'm working.
4. I am proud of the work I have delivered so far for our product.
5. I am learning things that I look forward to applying to future products.
6. My workstation seems to disappear out from under me while I'm working.
7. It's easy to get my workstation into the state I need to develop our product.
8. What aspect of our workstation setup is painful?
9. It's easy to run our software on my workstation while I’m developing it.
10. I can boot our software up into the state I need with minimal effort.
11. What aspect of running our software locally is painful? What could we do to make it less painful?
12. It's easy to run our test suites and to author new ones.
13. Tests are a stable, reliable, seamless part of my workflow.
14. Test failures give me the feedback I need on the code I am writing.
15. What aspect of production support is painful?
16. We collaborate well with the teams whose software we integrate with.
17. When necessary, it is within my power to request timely changes from other teams.
18. I have the resources I need to test and code confidently against other teams' integration points.
19. What aspect of integrating with other teams is painful?
20. I'm rarely impacted by breaking changes from other tracks of work.
21. We almost always catch broken tests and code before they're merged in.
22. What aspect of committing changes is painful?
23. Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
24. 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.
25. What aspect of our release process (CI/CD) is painful?
26. Our team releases new versions of our software as often as the business needs us to.
27. We are meeting our service-level agreements with a minimum of unplanned work.
28. 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.

Platform marketing, advocacy, consulting

Organizational Learning

Focus on ways of working…..


www.mercedes-benz.io/who-we-are/

Mercedes-Benz | Home | Who we are | How we work | What we do | Jobs | Blog

OWNERSHIP

Each of us is a responsible owner of multiple roles. We walk the talk and make things happen. We are masters of self-organization and let our entrepreneurial spirit run free. Failure is allowed and respected. We back each other up, share our experiences and find solutions together because the tribe is greater than the sum of its individual parts. We win and learn as one team.

PURPOSE

Our favorite question is 'why'. This is why every single circle and role in our organization follows a purpose, a clear north star, a shared goal to aim for. We keep our eyes on the prize and our desired outcome in the back of our mind to make sure we all follow a common mission and share the same beliefs for the bigger picture.

CHALLENGING

"Mission accomplished" is not enough. We reflect and challenge everything – the status quo, Mercedes-Benz and ourselves. We love taking on a good challenge and pushing boundaries where we see potential for evolution. Our focus always stays on what creates the highest sustainable impact and how to get there.

COLLABORATIVE

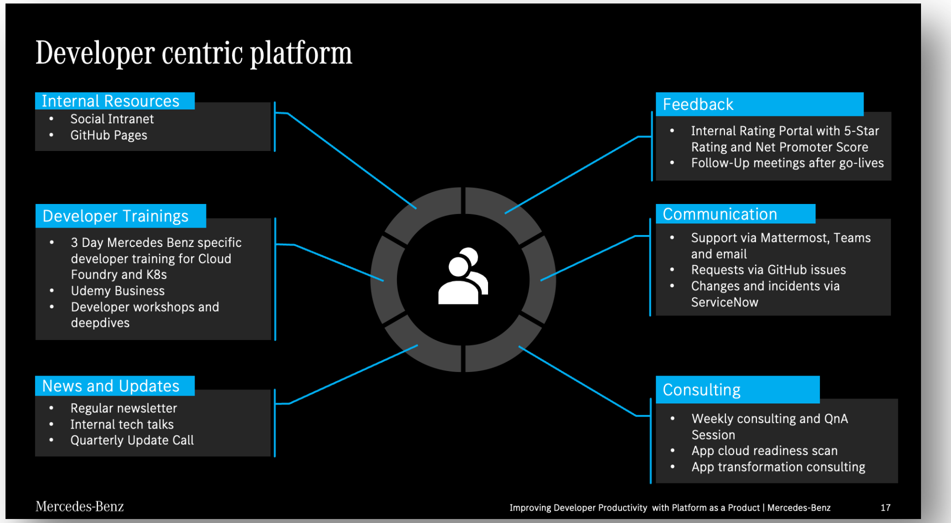
As a flat-hierarchical company we work as we live: collaborative, reflective, ambitious and united. We know there are more ways than one to reach targets. Each individual connection makes us

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

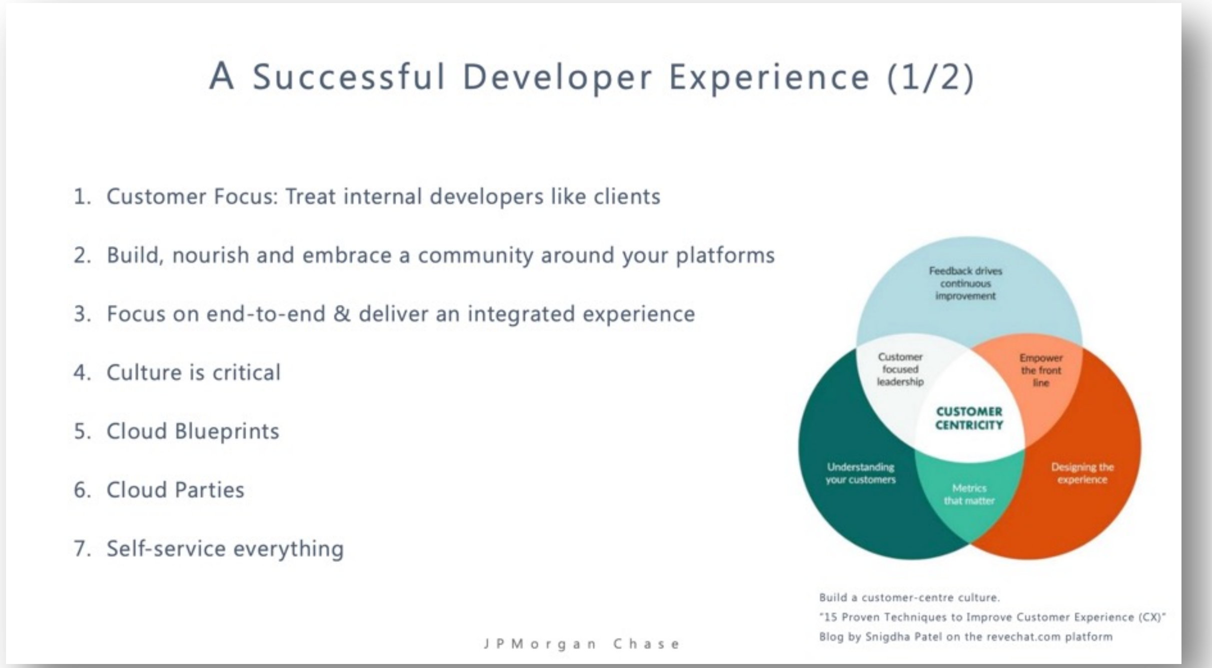
Consulting
- Weekly consulting and QnA Session
- App cloud readiness scan
- App transformation consulting

Mercedes-Benz

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

Sources: 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.

Platform lessons learned from 1,500+ application at JP Morgan Chase



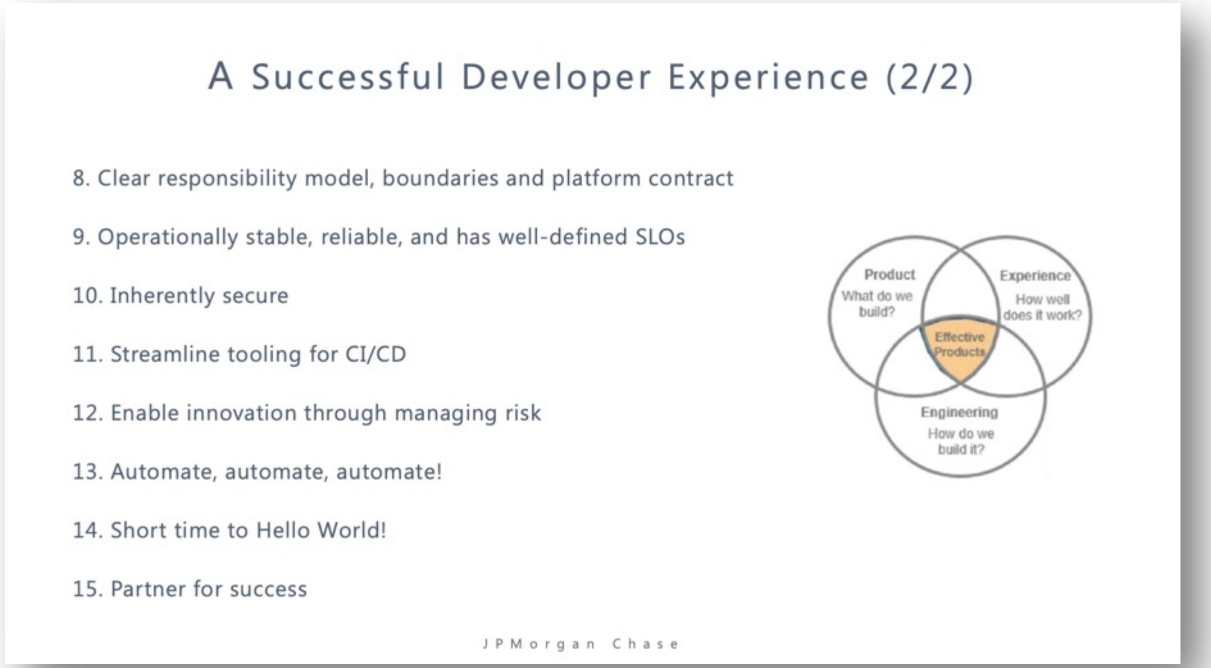
A Successful Developer Experience (1/2)

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

Feedback drives continuous improvement
Customer focused leadership
Empower the front line
CUSTOMER CENTRICITY
Understanding your customers
Designing the experience
Metrics that matter

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

JPMorgan Chase



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

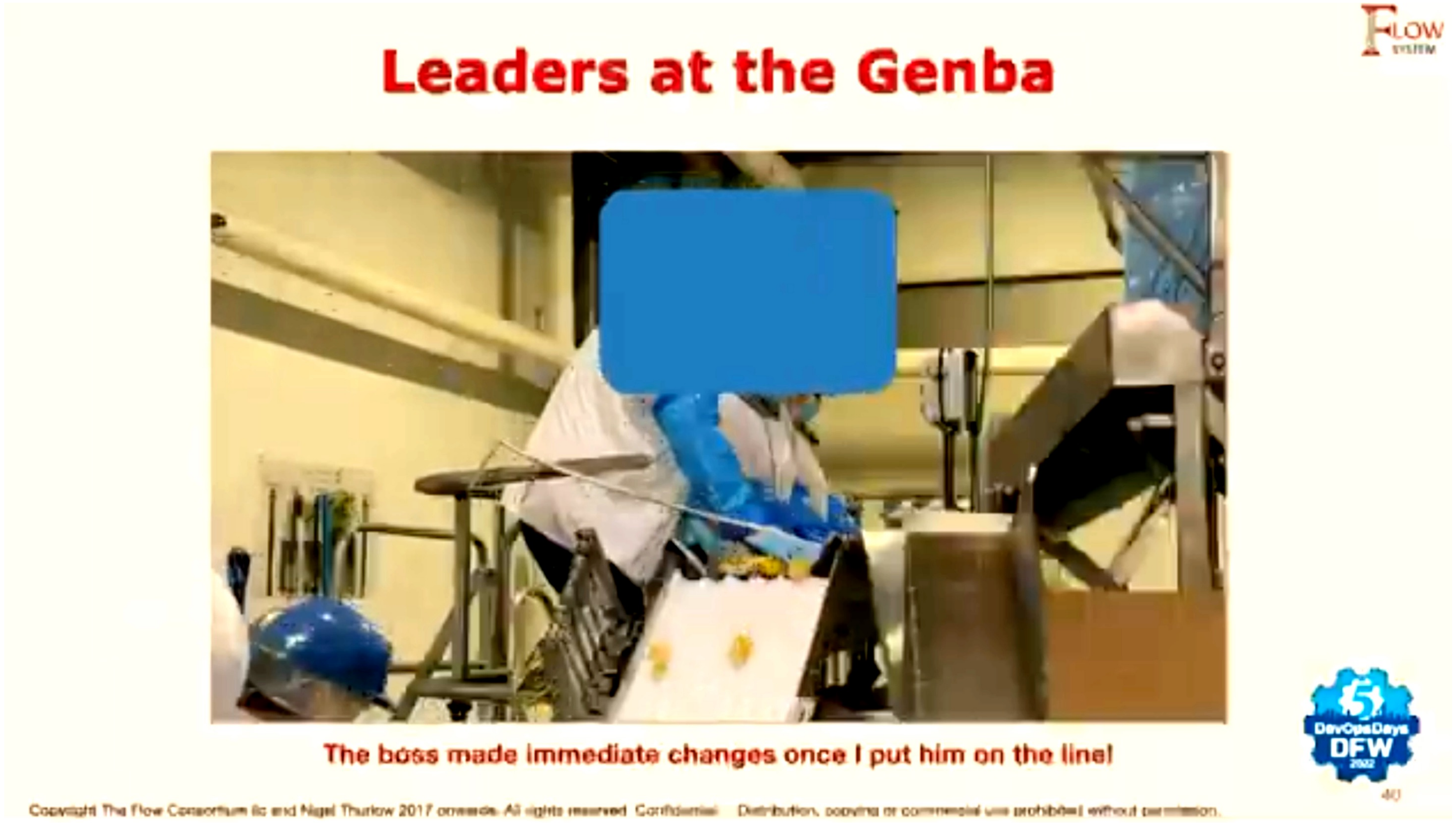
Product
What do we build?
Experience
How well does it work?
Effective Products
Engineering
How do we build it?

JPMorgan Chase

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

Management 👉 Culture

“Culture” means you



Leaders at the Genba

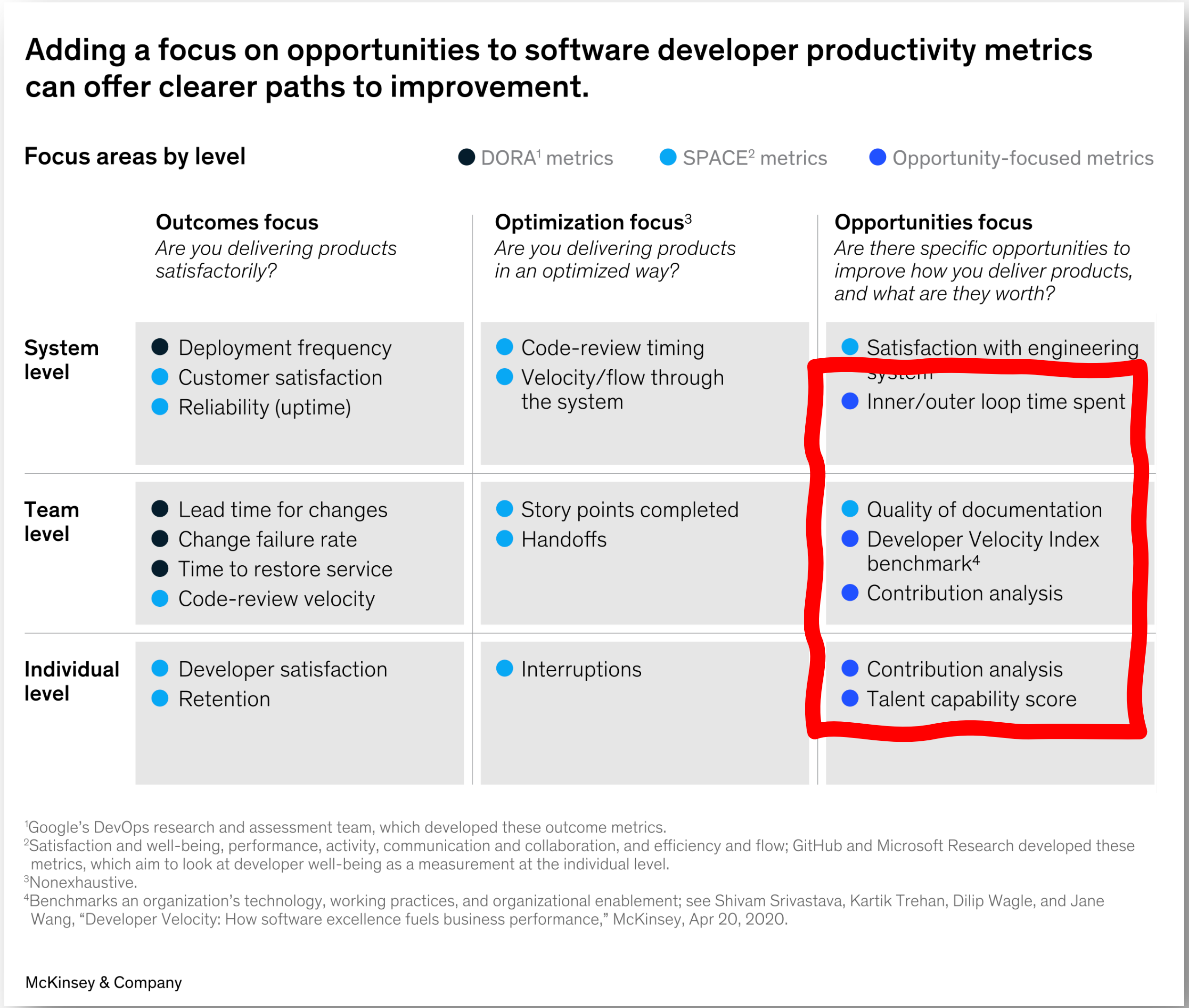
The boss made immediate changes once I put him on the line!

Copyright The Flow Consortium llc and Nigel Thurlow 2017 onwards. All rights reserved. Confidential. Distribution, copying or commercial use prohibited without permission.

40

Sources: “DevOps is Enterprise Wide,” Nigel Thurlow, DevOpsDays Dallas 2022.

Management likes dev productivity metrics, developers are skeptical



Adding a focus on opportunities to software developer productivity metrics can offer clearer paths to improvement.

Focus areas by level

⚫ DORA[1] metrics 🔵 SPACE[2] metrics 🟦 Opportunity-focused metrics

	Outcomes focus *Are you delivering products satisfactorily?*	**Optimization focus[3]** *Are you delivering products in an optimized way?*	**Opportunities focus** *Are there specific opportunities to improve how you deliver products, and what are they worth?*
System level	⚫ Deployment frequency 🔵 Customer satisfaction 🔵 Reliability (uptime)	🔵 Code-review timing 🔵 Velocity/flow through the system	🔵 Satisfaction with engineering system 🟦 Inner/outer loop time spent
Team level	⚫ Lead time for changes ⚫ Change failure rate ⚫ Time to restore service 🔵 Code-review velocity	🔵 Story points completed 🔵 Handoffs	🔵 Quality of documentation 🟦 Developer Velocity Index benchmark[4] 🟦 Contribution analysis
Individual level	🔵 Developer satisfaction 🔵 Retention	🔵 Interruptions	🟦 Contribution analysis 🟦 Talent capability score

[1]Google's DevOps research and assessment team, which developed these outcome metrics.
[2]Satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow; GitHub and Microsoft Research developed these metrics, which aim to look at developer well-being as a measurement at the individual level.
[3]Nonexhaustive.
[4]Benchmarks an organization's technology, working practices, and organizational enablement; see Shivam Srivastava, Kartik Trehan, Dilip Wagle, and Jane Wang, "Developer Velocity: How software excellence fuels business performance," McKinsey, Apr 20, 2020.

McKinsey & Company

Sources: "Yes, you can measure software developer productivity," many at, McKinsey, August, 2023. "The SPACE of Developer Productivity," March, 2021 . See also further commentary from Coté.

A thriving organization focuses on satisfaction, flow, ease, happiness

Causes of thriving	Because a developer is…
Agency	1) able to voice disagreement with team definitions of success 2) has a voice in how their contributions are measured
Motivation & Self-Efficacy	1) motivated when working on code at work 2) can see tangible progress most of the time 3) is working on the type of code work they want to work on 4) is confident that even when working in code is unexpectedly difficult, they will solve their problems
Learning Culture	1) learning new skills as a developer 2) able to share the things they learn at work
Support & Belonging	1) supported to grow, learn, and make mistakes by their team 2) agrees they are accepted for who they are by their team

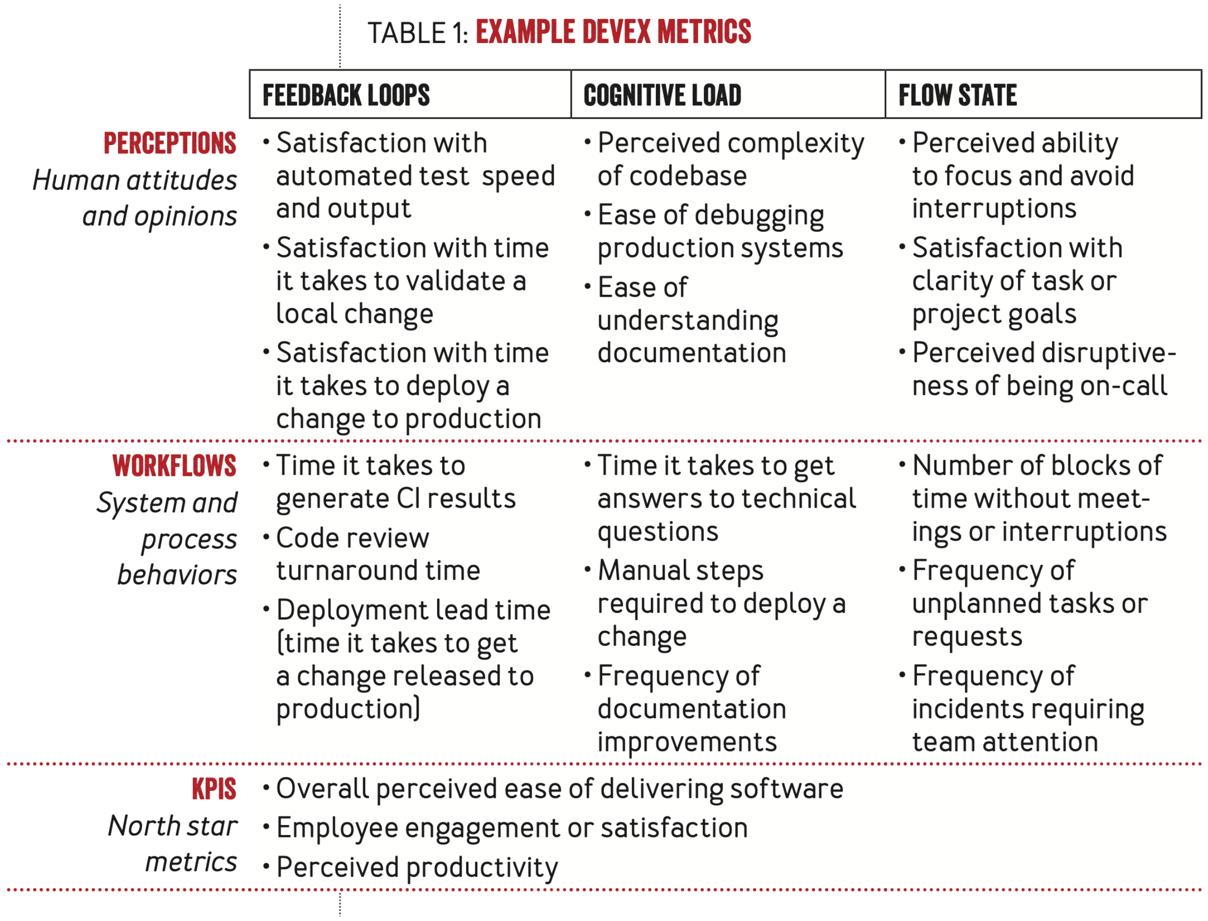


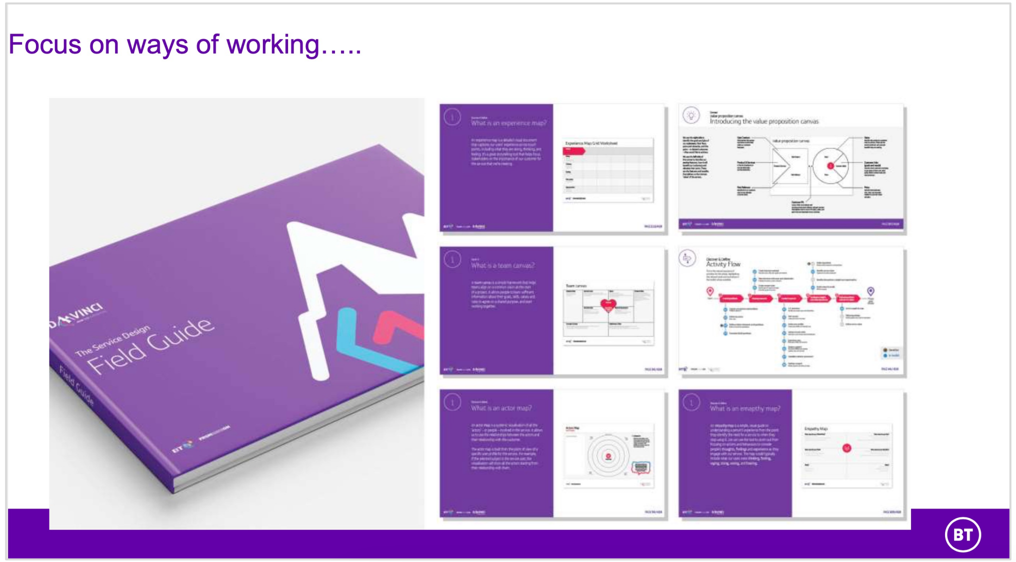
TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS *Human attitudes and opinions*	• Satisfaction with automated test speed and output • Satisfaction with time it takes to validate a local change • Satisfaction with time it takes to deploy a change to production	• Perceived complexity of codebase • Ease of debugging production systems • Ease of understanding documentation	• Perceived ability to focus and avoid interruptions • Satisfaction with clarity of task or project goals • Perceived disruptive-ness of being on-call
WORKFLOWS *System and process behaviors*	• Time it takes to generate CI results • Code review turnaround time • Deployment lead time (time it takes to get a change released to production)	• Time it takes to get answers to technical questions • Manual steps required to deploy a change • Frequency of documentation improvements	• Number of blocks of time without meet-ings or interruptions • Frequency of unplanned tasks or requests • Frequency of incidents requiring team attention
KPIS *North star metrics*	• Overall perceived ease of delivering software • Employee engagement or satisfaction • Perceived productivity		

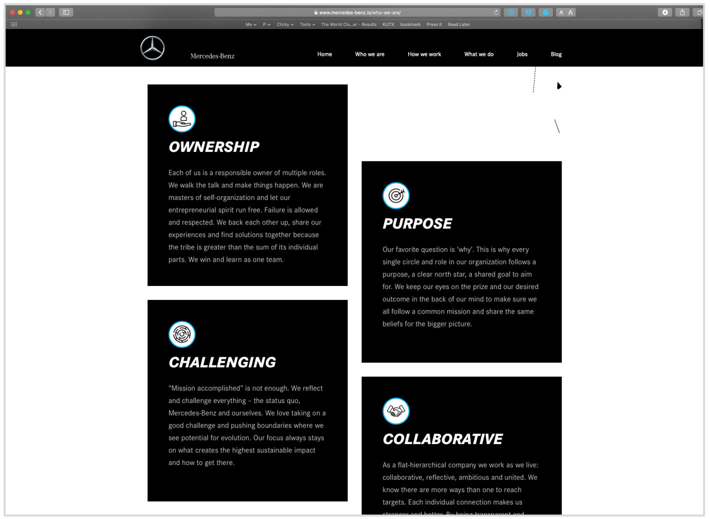
Sources: "Developer Thriving: The four factors that drive Software Developer Productivity across Industries," March, 2023; "DevEx: What Actually Drives Productivity," Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Michaela Greiler, April 2023; "DevEx in Action: A study of its tangible impacts," Dec 2023.

Platform marketing, advocacy, consulting

Organizational Learning



Focus on ways of working…..



www.mercedes-benz.io/who-we-are/

OWNERSHIP

Each of us is a responsible owner of multiple roles. We walk the talk and make things happen. We are masters of self-organization and let our entrepreneurial spirit run free. Failure is allowed and respected. We back each other up, share our experiences and find solutions together because the tribe is greater than the sum of its individual parts. We win and learn as one team.

PURPOSE

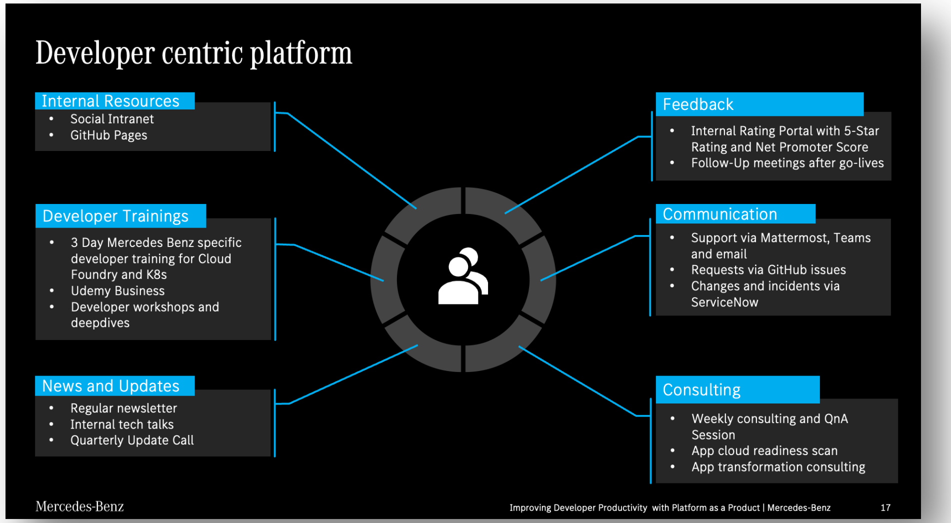
Our favorite question is 'why'. This is why every single circle and role in our organization follows a purpose, a clear north star, a shared goal to aim for. We keep our eyes on the prize and our desired outcome in the back of our mind to make sure we all follow a common mission and share the same beliefs for the bigger picture.

CHALLENGING

"Mission accomplished" is not enough. We reflect and challenge everything – the status quo, Mercedes-Benz and ourselves. We love taking on a good challenge and pushing boundaries where we see potential for evolution. Our focus always stays on what creates the highest sustainable impact and how to get there.

COLLABORATIVE

As a flat-hierarchical company we work as we live: collaborative, reflective, ambitious and united. We know there are more ways than one to reach targets. Each individual connection makes us



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

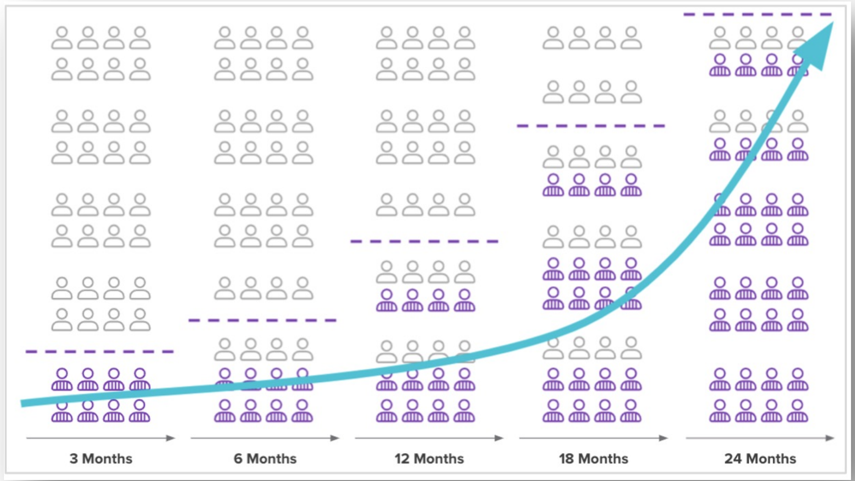
Communication
- Support via Mattermost, Teams and email
- Requests via GitHub issues
- Changes and incidents via ServiceNow

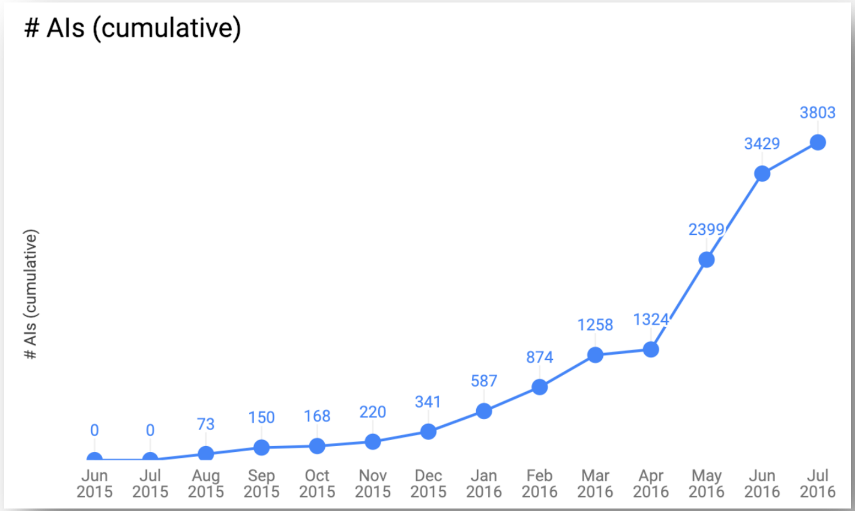
Consulting
- Weekly consulting and QnA Session
- App cloud readiness scan
- App transformation consulting

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

Sources: 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.

Scaling Phase – Pairing & Seeding to build trust & training



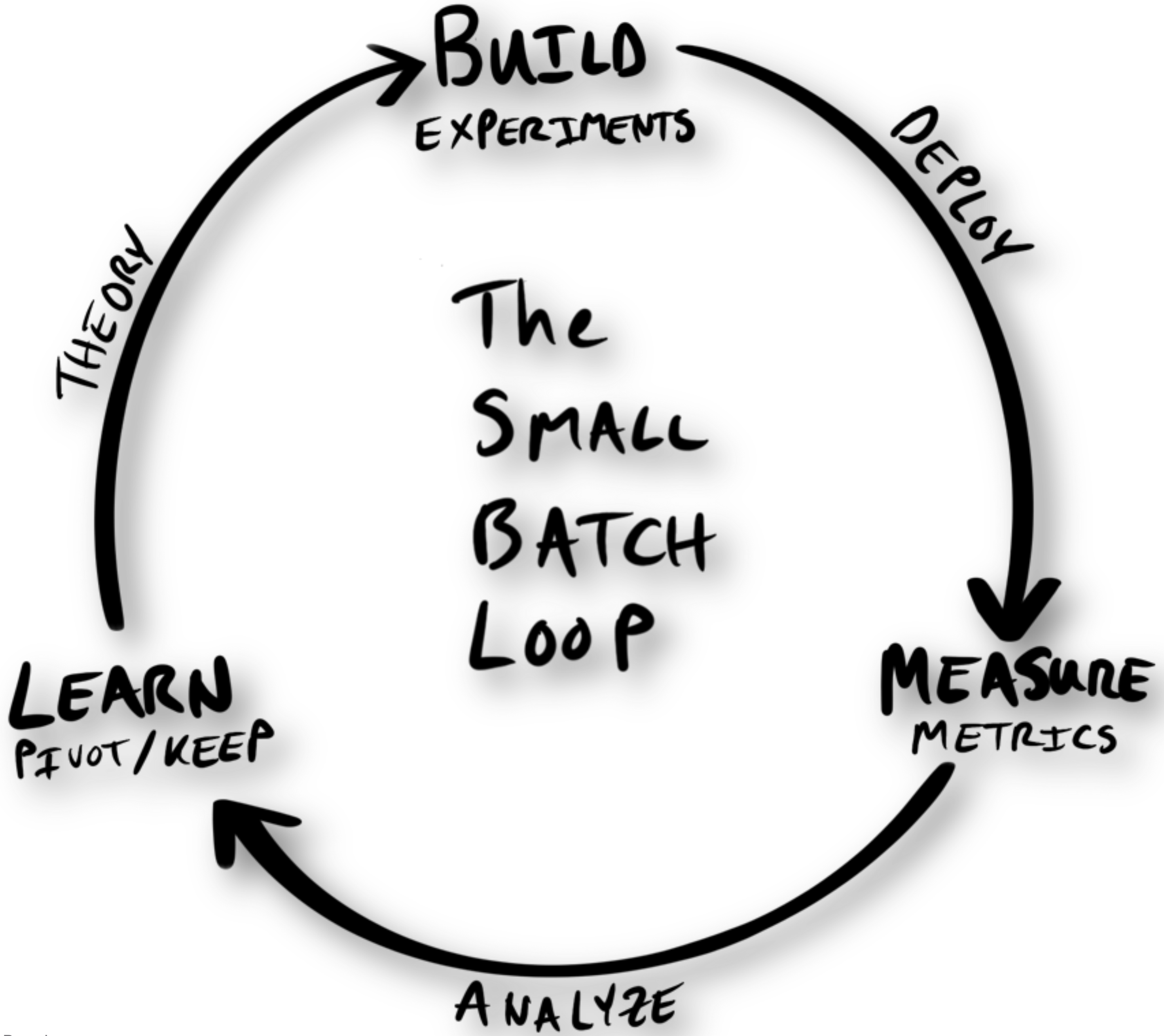


1. Create platform team.
2. Pick one or two apps, real apps.
3. Develop the apps & platform together.
4. Do this for three months.
5. Pick some more apps, to taste.
6. Seed app people to new teams.
7. GOTO 3.

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.

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, Oct 2023.


BUILD
EXPERIMENTS
DEPLOY
MEASURE
METRICS
ANALYZE
LEARN
PIVOT / KEEP
THEORY
The Small Batch Loop

All too common challenges & blockers

People, Culture, etc.

- Skills, hiring
- Reluctance to change
- Scaling new roles
- Org. structure
- “We already do agile.”
- Durability through people & org. change

Planning & Alignment

- Budgeting
- Misaligned executives
- IT is still in the basement
- Compliance
- Weak connection to business value

Technical Execution

- ITSM instead of Platforms
- Overwhelming legacy portfolio
- Dependencies between teams
- Local optimization, no CI/CD

Thanks!

🏢 cote@broadcom.com

📚 https://cote.io