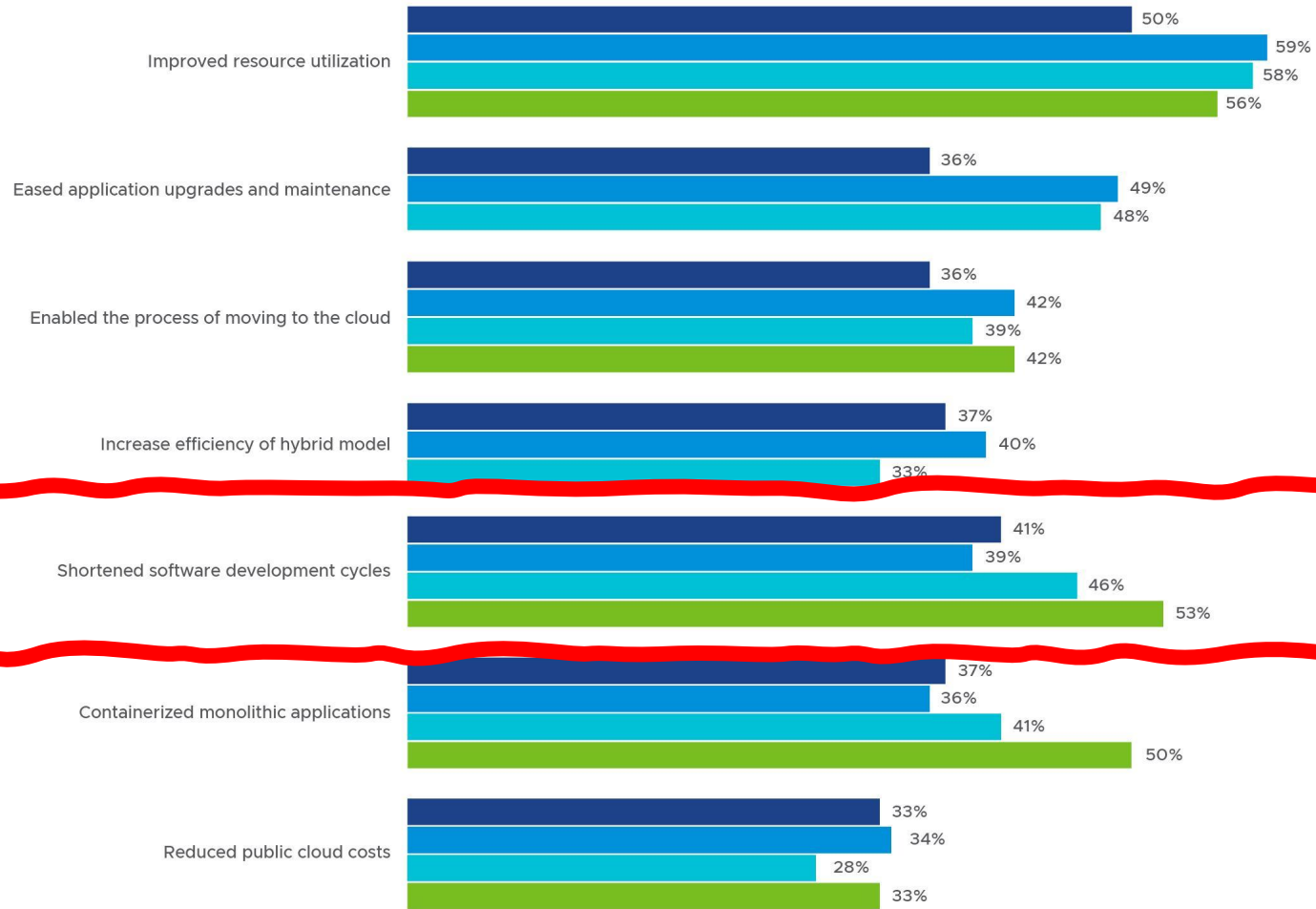


7 Lessons from 7 Years of Running Platforms

Coté

June, 2023

What benefits has your organization realized from operating Kubernetes?
(Choose all that apply)



Source: State of Kubernetes 2022, UMuar

■ 2020 ■ 2021 ■ 2022 ■ 2023

Hello, I am Coté.

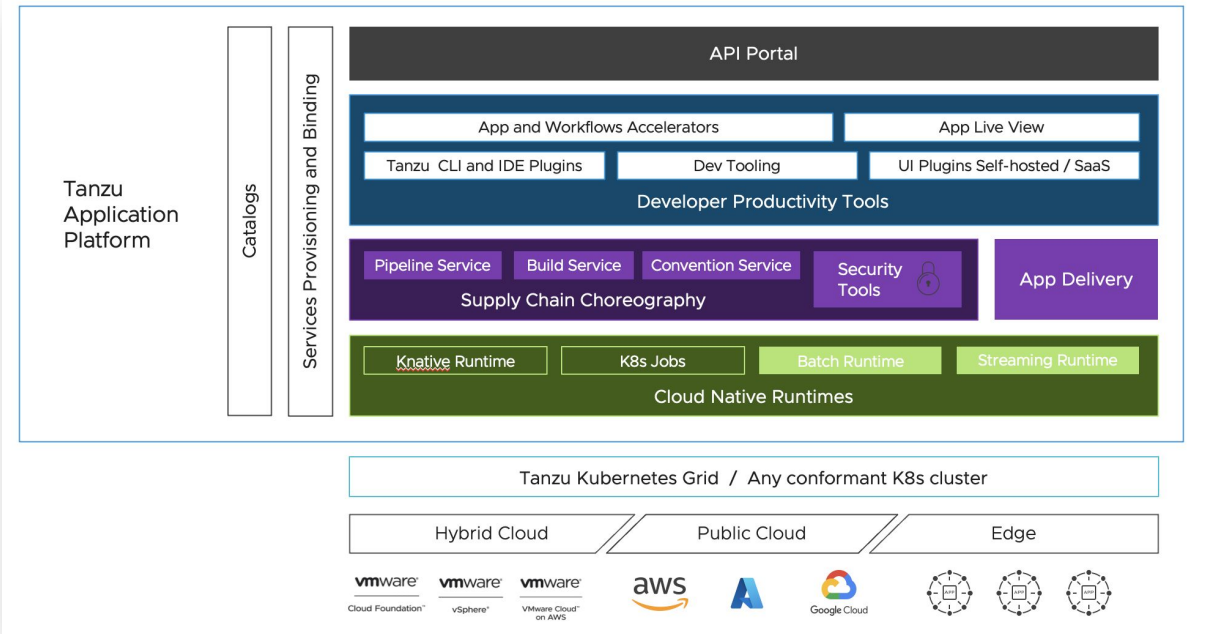
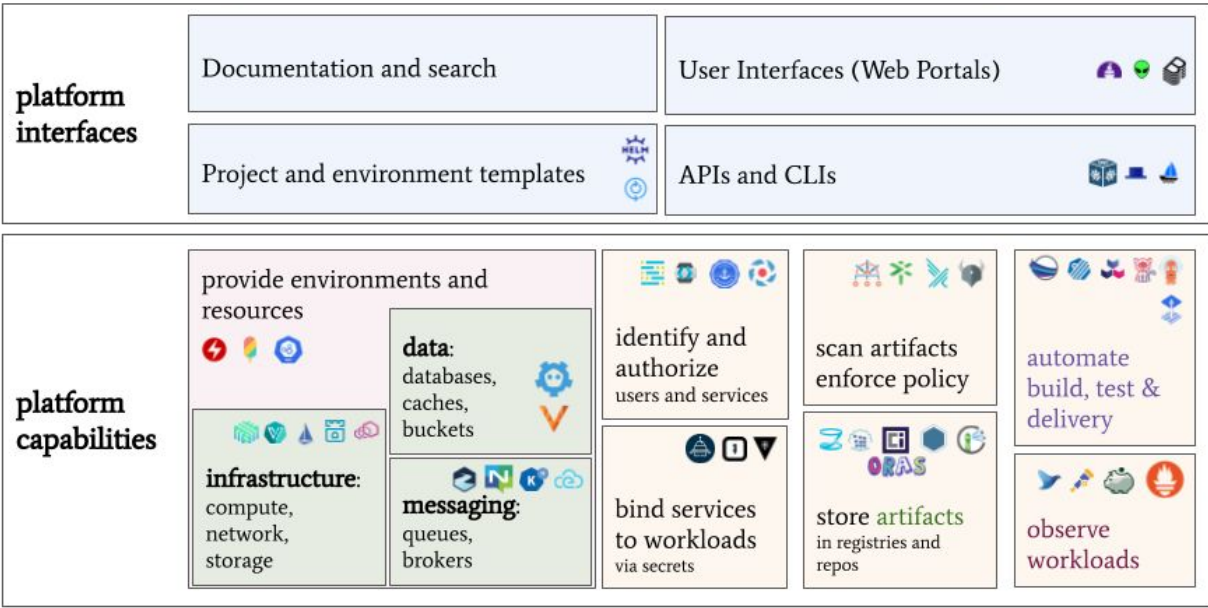
I work at **vmware**[®]

More: <https://cote.io>



Research & Development
FY05 - Team Innovation
Michael Cote

Product and application teams



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

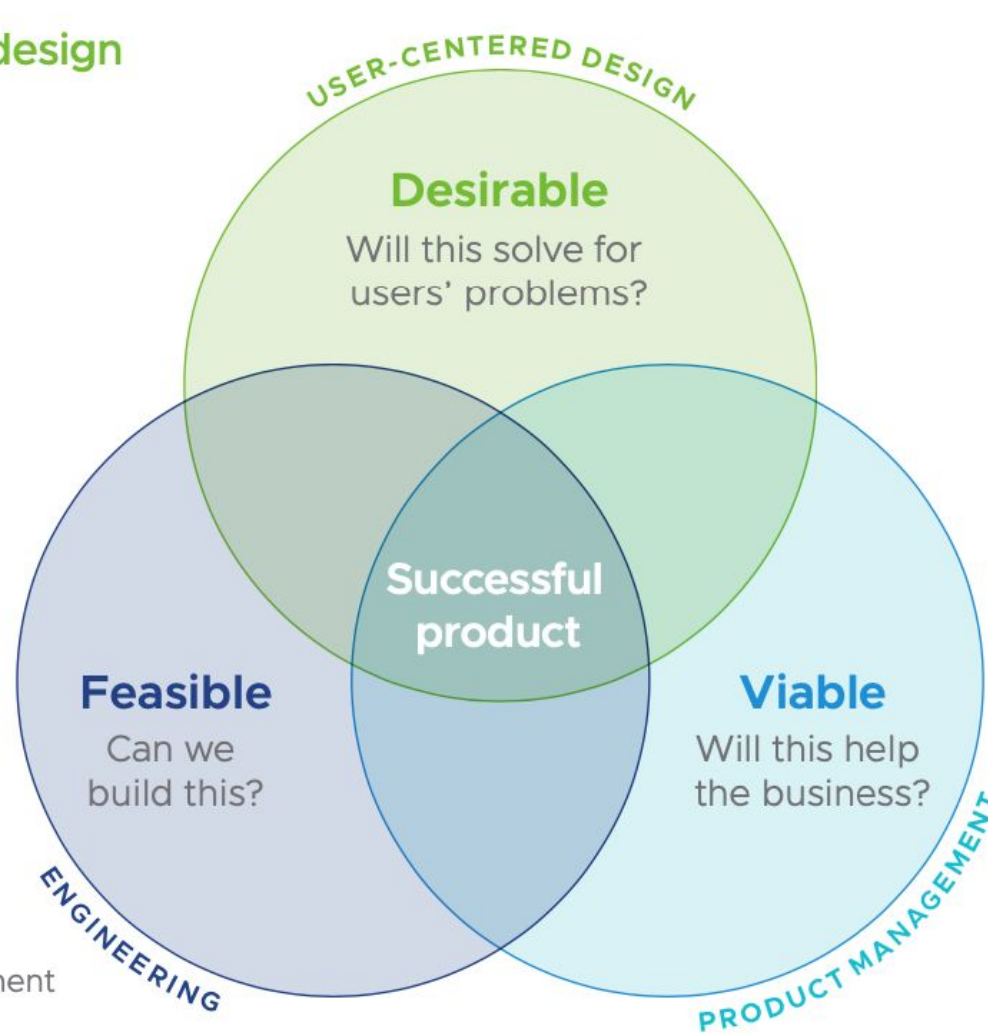
Thomas Müller, Mercedes-Benz



User-centered design

ACTIVITIES:

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



Agile / XP

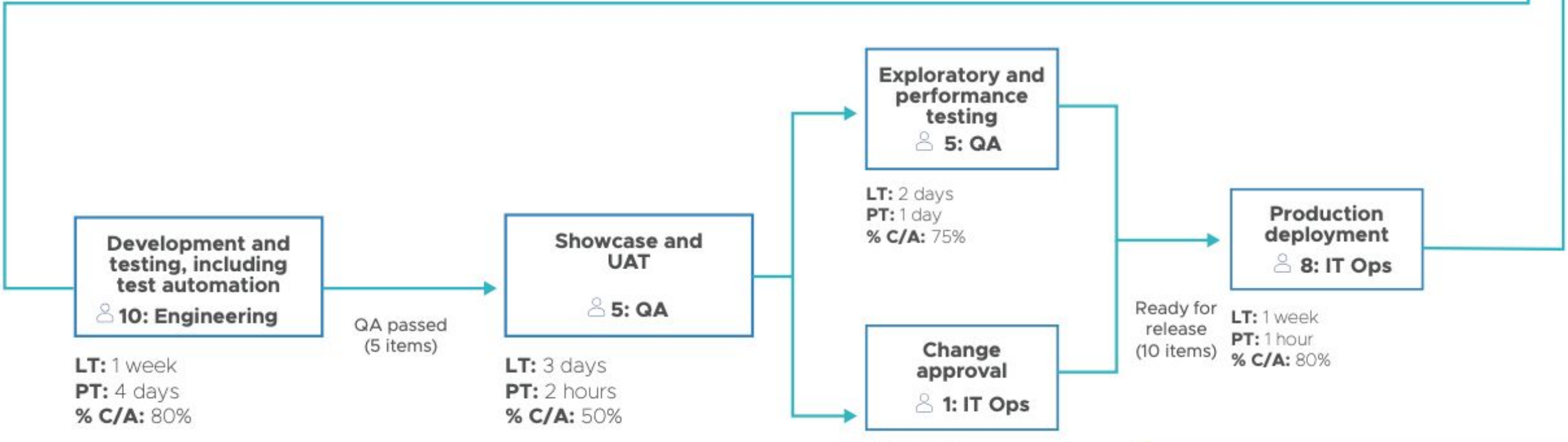
ACTIVITIES:

- 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

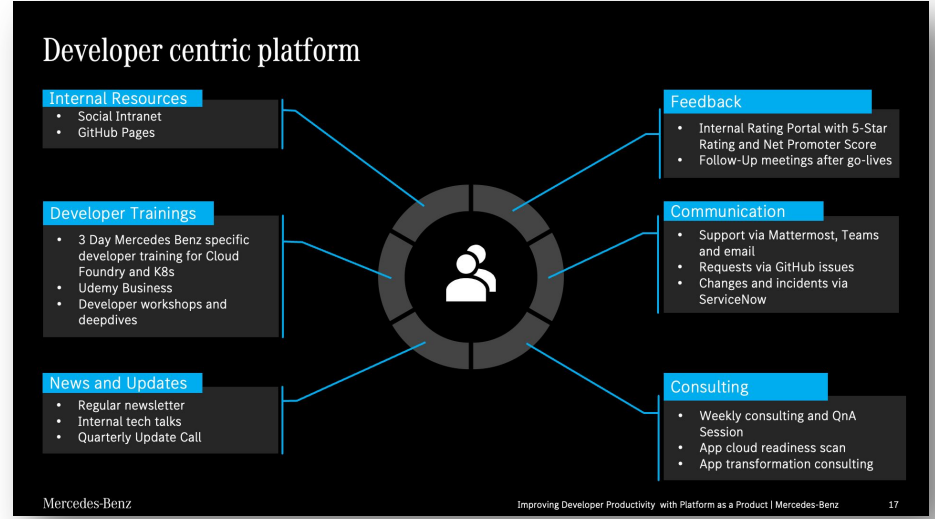
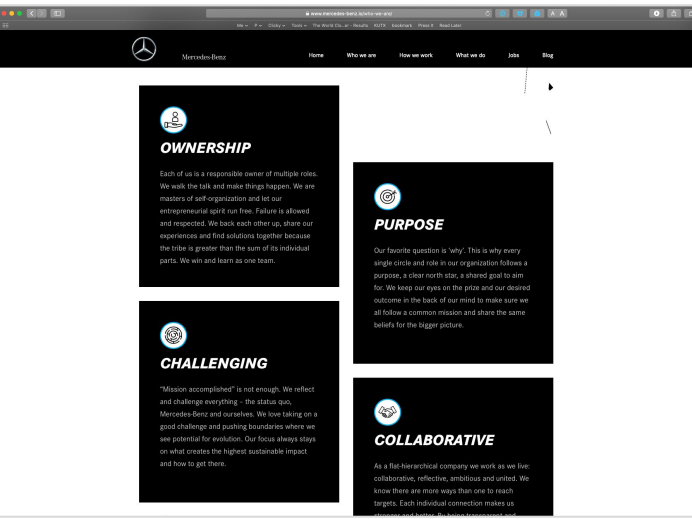


Total Lead Time (LT): 9.5 weeks
Total Processing Time (PT): 7 days
Activity ratio: 14.5%
Rolled % Complete & Accurate(C/A): 5.4%

Find the 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.
- 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.

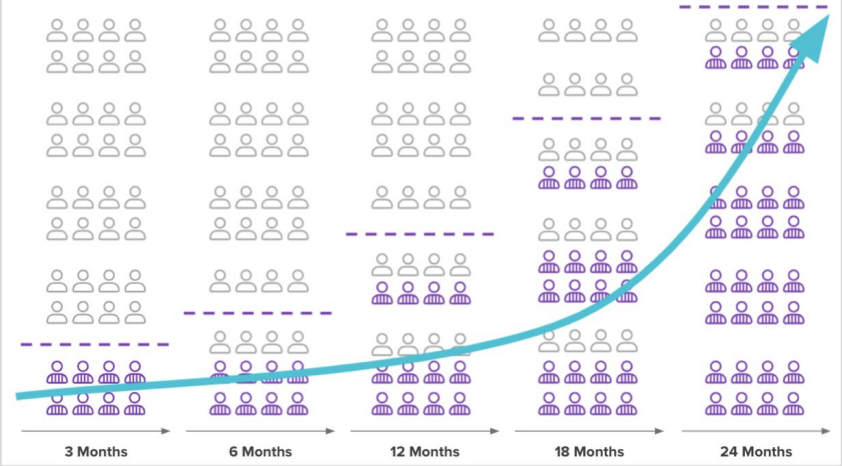
Platform marketing, advocacy, consulting



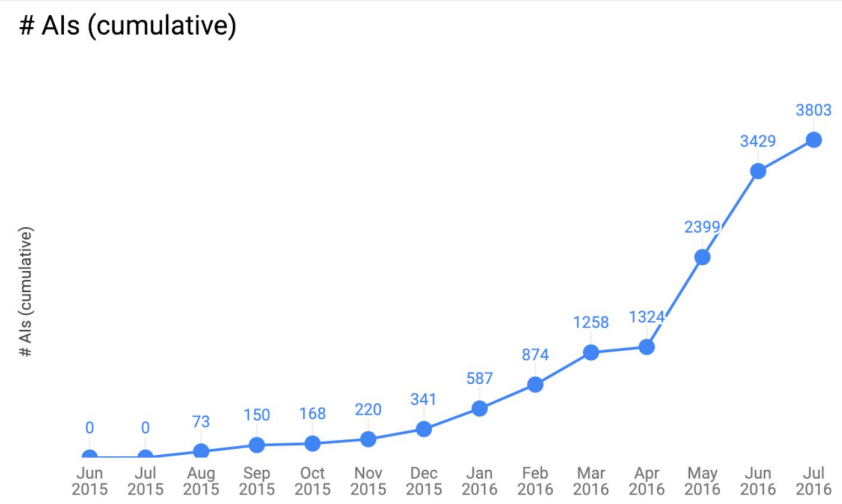
Initial Phase – Create the Platform with Developers

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

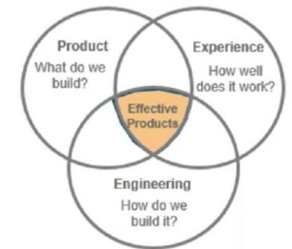


Build a customer-centre culture.
"15 Proven Techniques to Improve Customer Experience (CX)"
Blog by Snigdha Patel on the revechat.com platform

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



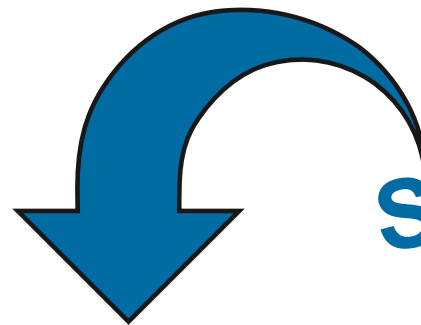
Thanks!

 <https://newsletter.cote.io>

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

 cotem@vmware.com

 [@cote@social.io](https://social.io)



Slides & stuff

