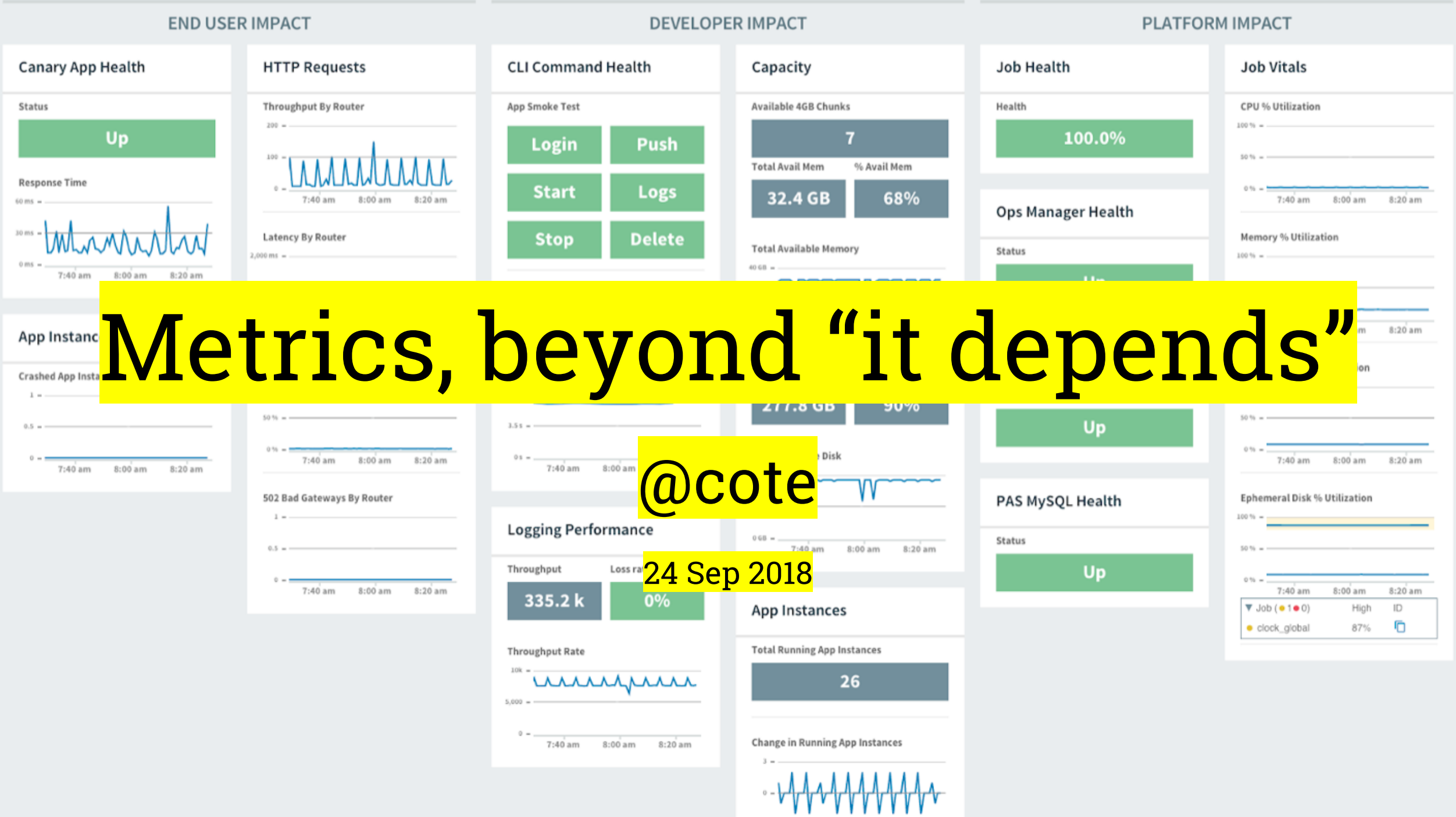
Metrics, beyond “it depends”

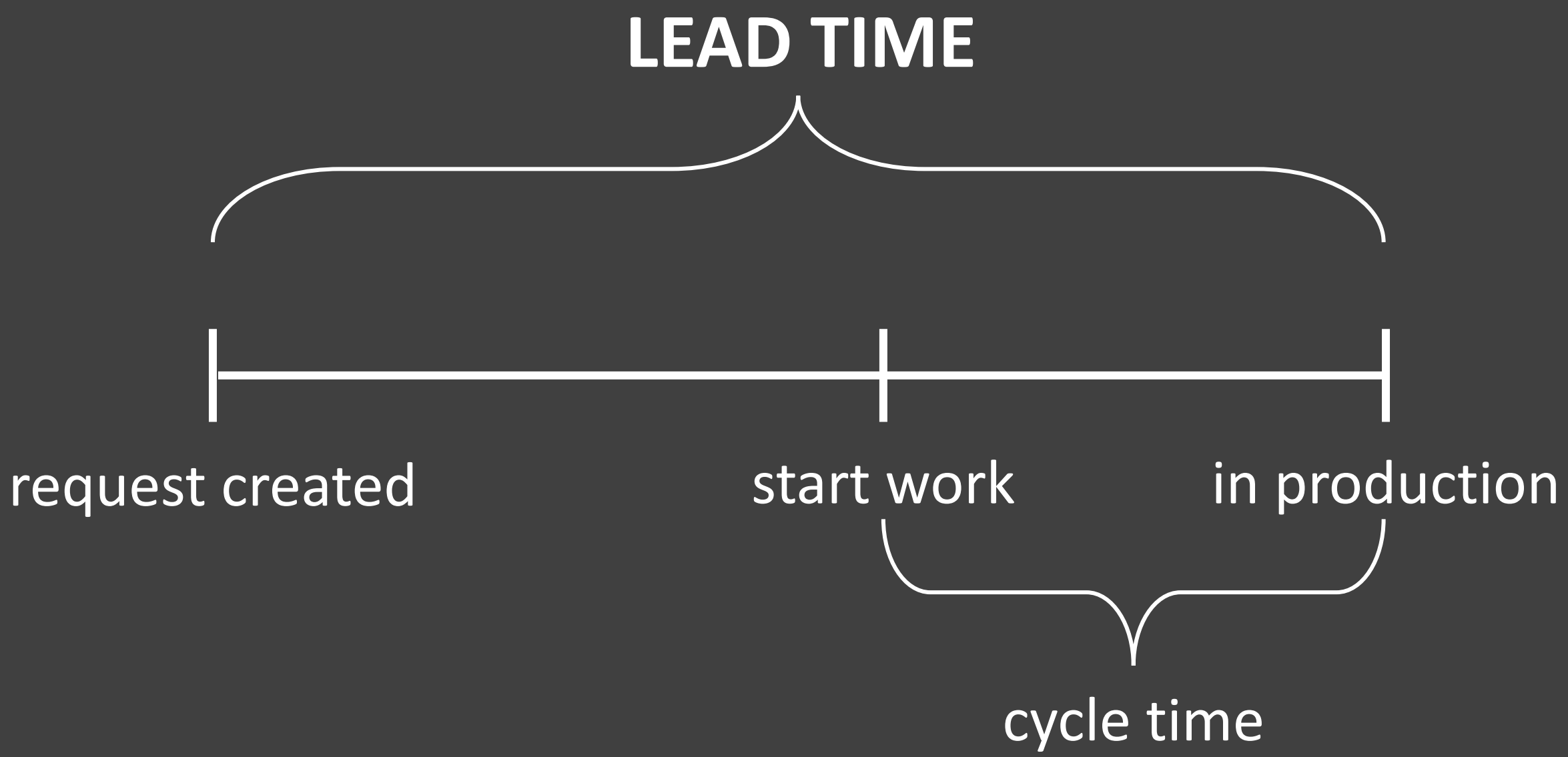
@cote

24 Sep 2018



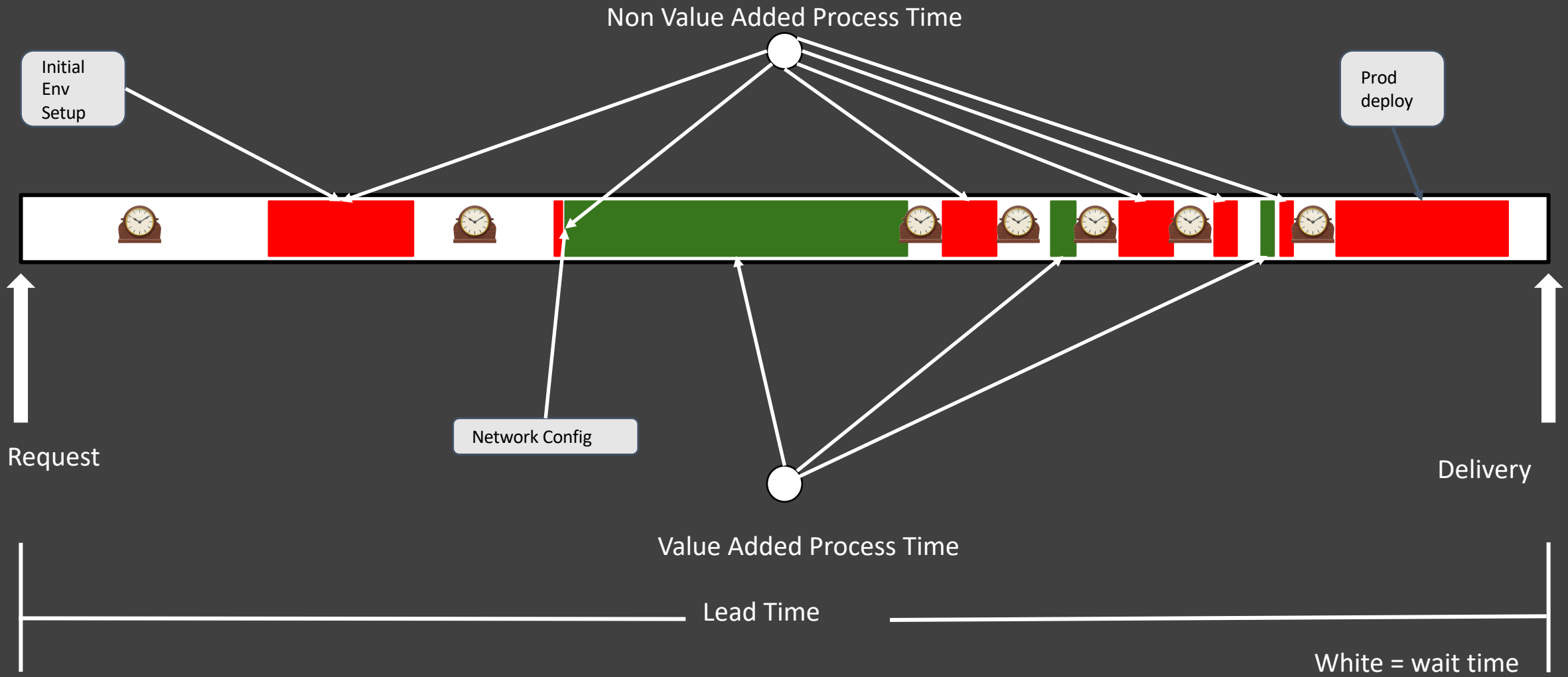
Why?

- Optimize – speed, better design
- Monitoring – find problems
- Be useful – helping the business



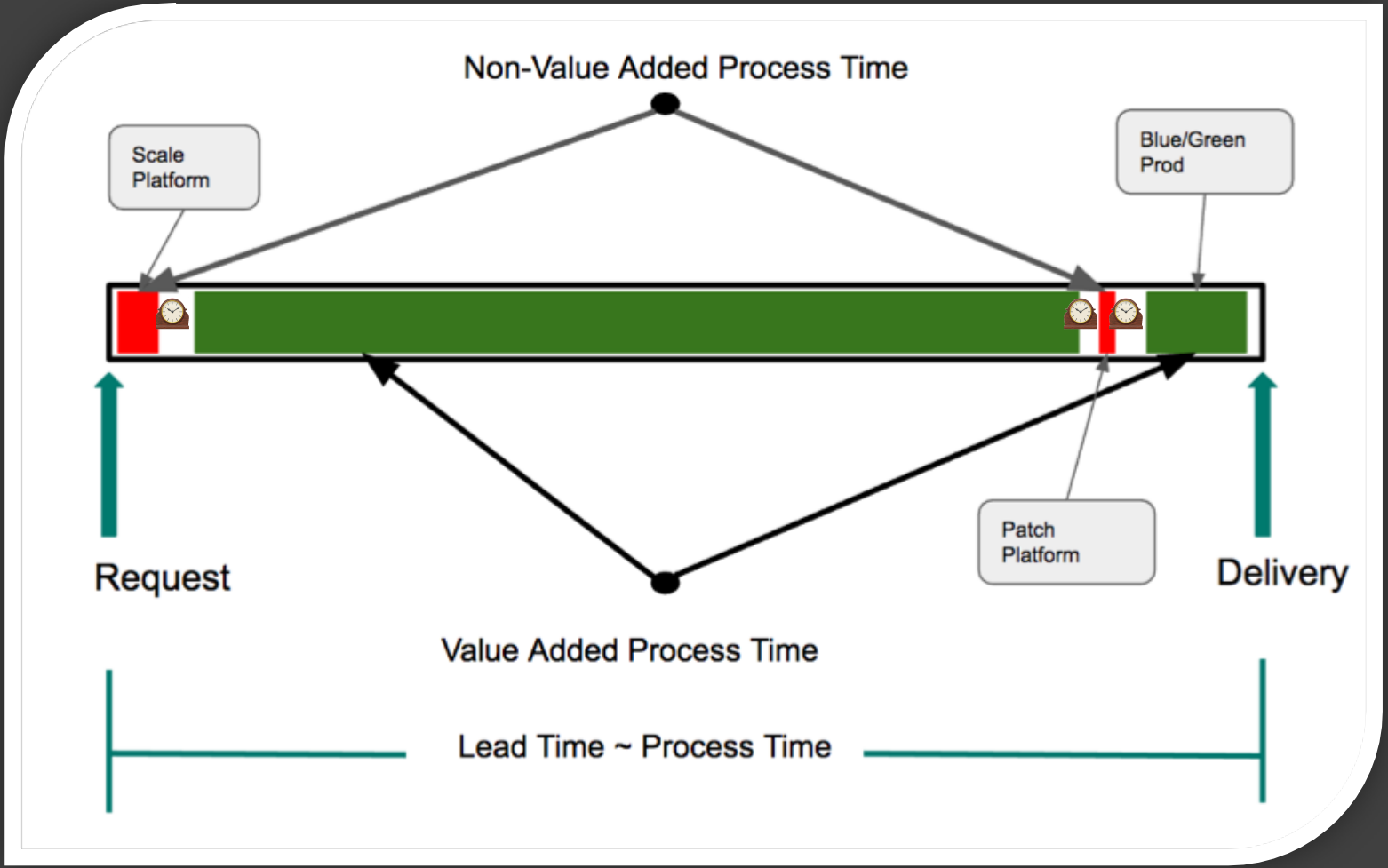
Source: modified from “Kanban: Definition of Lead Time and Cycle Time,” Stefan Roock, March, 2010.

Example 6 week lead time



Source: scenario based on Pivotal customers; "Crossing the Value Stream: Improving Development with Pivotal and Cloud Foundry," Matt Gunter, Pivotal, 2018.

Better value stream, 3-6 weeks lead time



Apply ruthless automation

Remove "bullshit" work

~85% reduction in ops time

~50% reduction in release timeline

Source: scenario based on Pivotal customers; "Crossing the Value Stream: Improving Development with Pivotal and Cloud Foundry," Matt Gunter, Pivotal, 2018.

Uses

- Drives focus, trackable
- Dumb stuff discovery
- Silo breaking

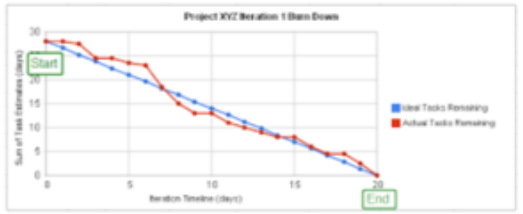
Halve cycles for more feedback

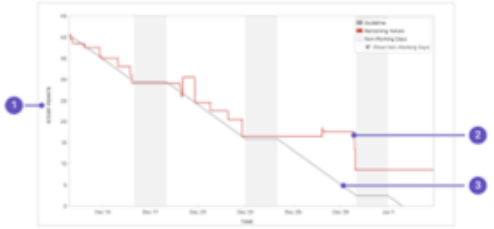
or

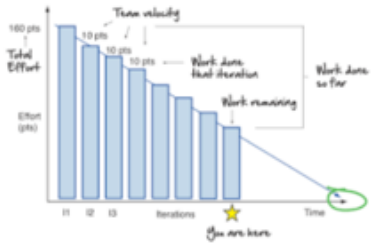
Deliver more in each cycle

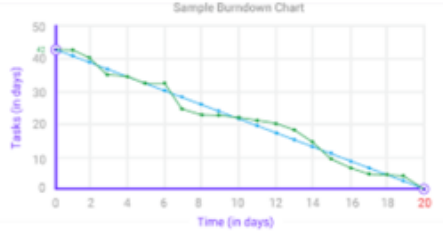
or

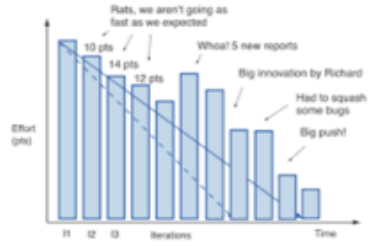
Don’t tell no one, work less

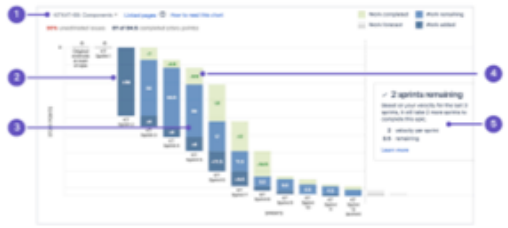


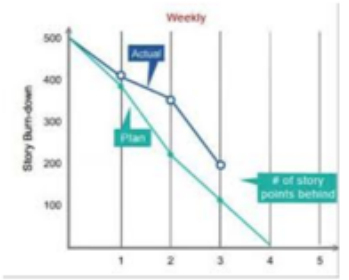


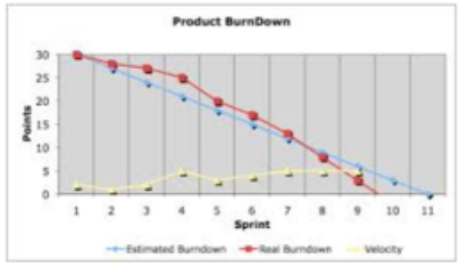


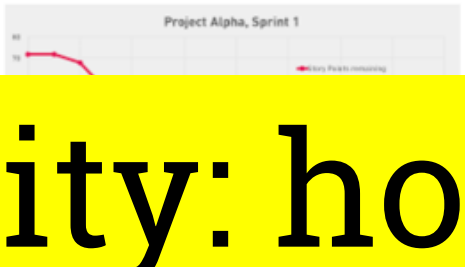


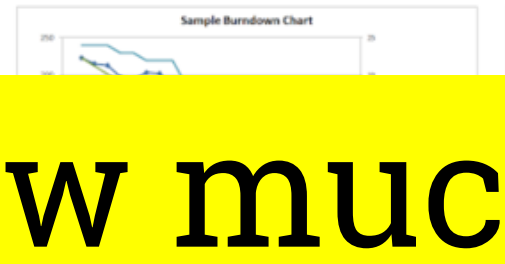


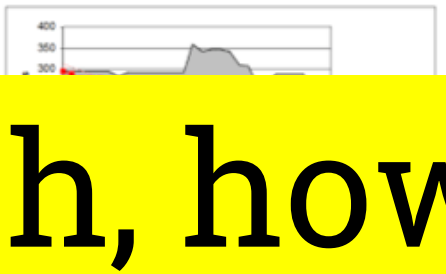




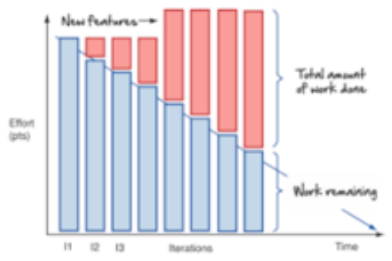


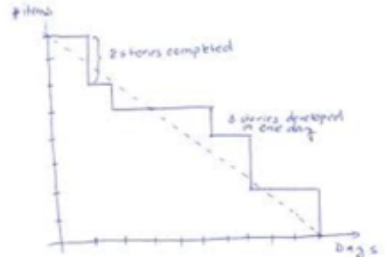


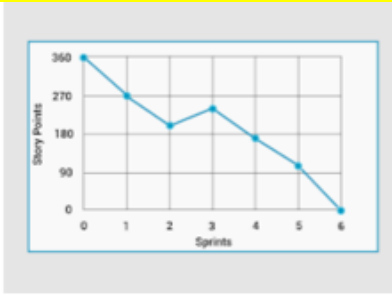


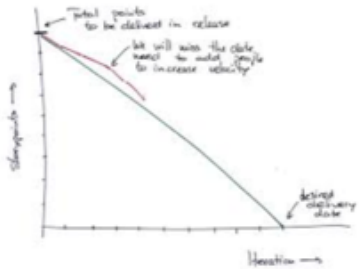


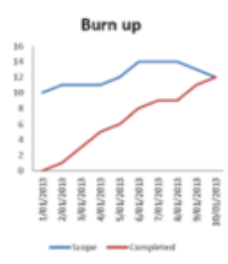
Velocity: how much, how often

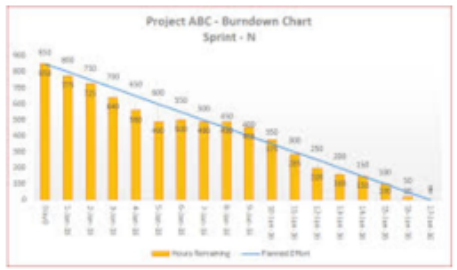












Uses

- Planning
- Predictability

Measure your **backlog** to keep it small

Source: photo by Jeser Bröring.

Latency: response time, click-to-screen

“53% of mobile users abandon sites that take longer than 3 seconds to load”

Source: "Mobile Load Time and User Abandonment," Tammy Everts, Akamai, Sep, 2016.

Uses

- Happy user, happy developer
- Easily find obvious problems

All the great hipster metrics

- **Google SRE:**

  Latency, Traffic, Errors, and Saturation

- **USE Method:**

  Utilization, Saturation, and Errors

- **RED Method:**

  Rate, Errors, and Duration

Source: ”How to Monitor the SRE Golden Signals,” Steve Mushero, Nov, 2017.


“The Business” (mostly money)

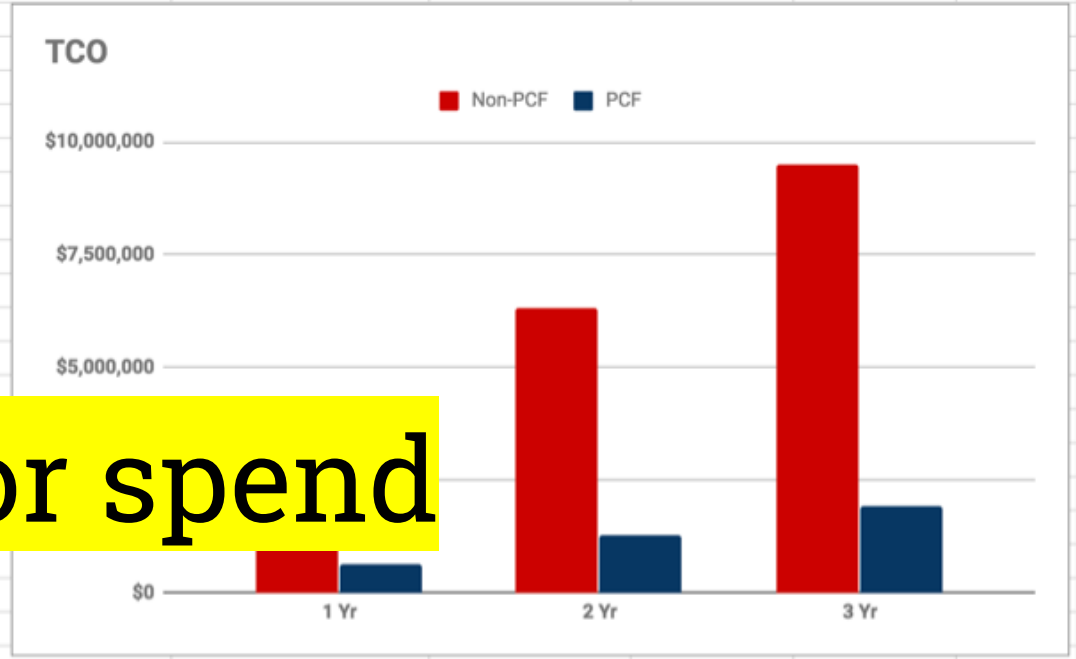
ROI ?

IT'S A TRAP!

fx Coding

	A	B	C	D	E
1	Coding				
2	Initial Environment Setup before PCF	600.00			
3	Initial Environment Setup after PCF	480.00		**Initial Env Setup**	80.00%
4					
5	Time Spent Coding before PCF	9.50			
6	Time Spent Coding after PCF	22.00		**Developer Productivity Gains**	231.58%
7					
8	Day 2 Ops Processing Time before PCF	583.08			
9	Day 2 Ops Processing Time after PCF	120.00		**Day 2 Operations Improvement**	79.42%
10					
11	Annual Release PT before PCF	57.00			
12	Annual Release PT after PCF	3.50		**RA Productivity Improvement**	93.86%
13					
14					
15	Number of existing traditional apps				
16	Hourly rate				
17	Working Weeks per year				
18	Number of developers				
19	# Apps Per Year Requiring New Env Setup				
20					
21	Annual Initial Env Setup - Traditional	$60,600			
22	Annual Initial Env Setup - PCF	$48,480			$17,064,960
23	Annual Initial Env Setup Savings	$12,120			$7,368,960
24					
25	Annual Development Productivity Savings	$9,696,000	<-- These savings are not rolled into TCO		
26					
27	Annual Day 2 Ops - Traditional	$2,826,757		$1,880,000	
28	Annual Day 2 Ops - PCF	$581,760			
29	Annual Day 2 Ops Savings	$1,298,240			
30					
31	Annual Release Cost - Traditional	$276,336			
32	Annual Release Automation Cost - PCF	$16,968			
33	Annual Release Cost Savings	$259,368			
34					
35	Annual Infrastructure Savings	$2,164,000		<-- These savings are not rolled into TCO	
36					
37	Annual Security Savings	$0		<-- These savings are not rolled into TCO	
38					

Focus on value for spend
(and also less spend)



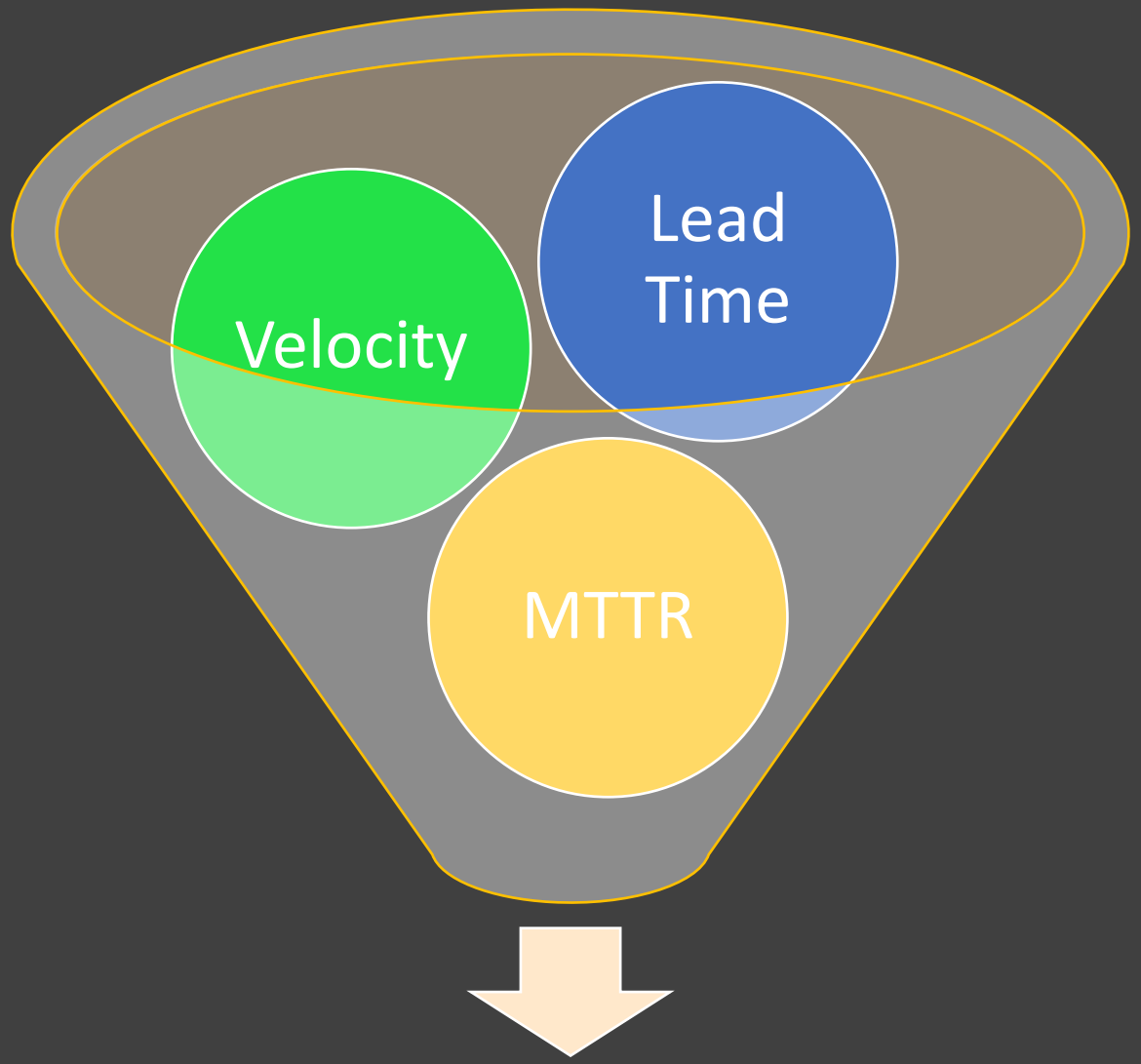
	Traditional Non-Dev Costs	PCF Ops Costs	Yearly Savings
	Non-PCF	PCF	
1 Yr	$3,163,693	$647,208	$2,516,485
2 Yr	$6,327,386	$1,294,416	$5,032,970
3 Yr	$9,491,079	$1,941,624	$7,549,455
		3 year TCO Savings	$15,098,910

Source: Pivotal Platform Architect organization.

f Measure - Inputs | Speed - Initial Env Setup | Speed - Release Automation | Speed - App Dev | Speed - Day 2 Ops | Savings - Infrastructure | Security | Summary | Actor

HA HA

SMARTART



TIME TO MARKET

(Get money faster, grab marketshare before competition)

tankerplanning

(Theoretic) opportunity cost:
If we stay the same, how much money we will not make (or, “lose”)

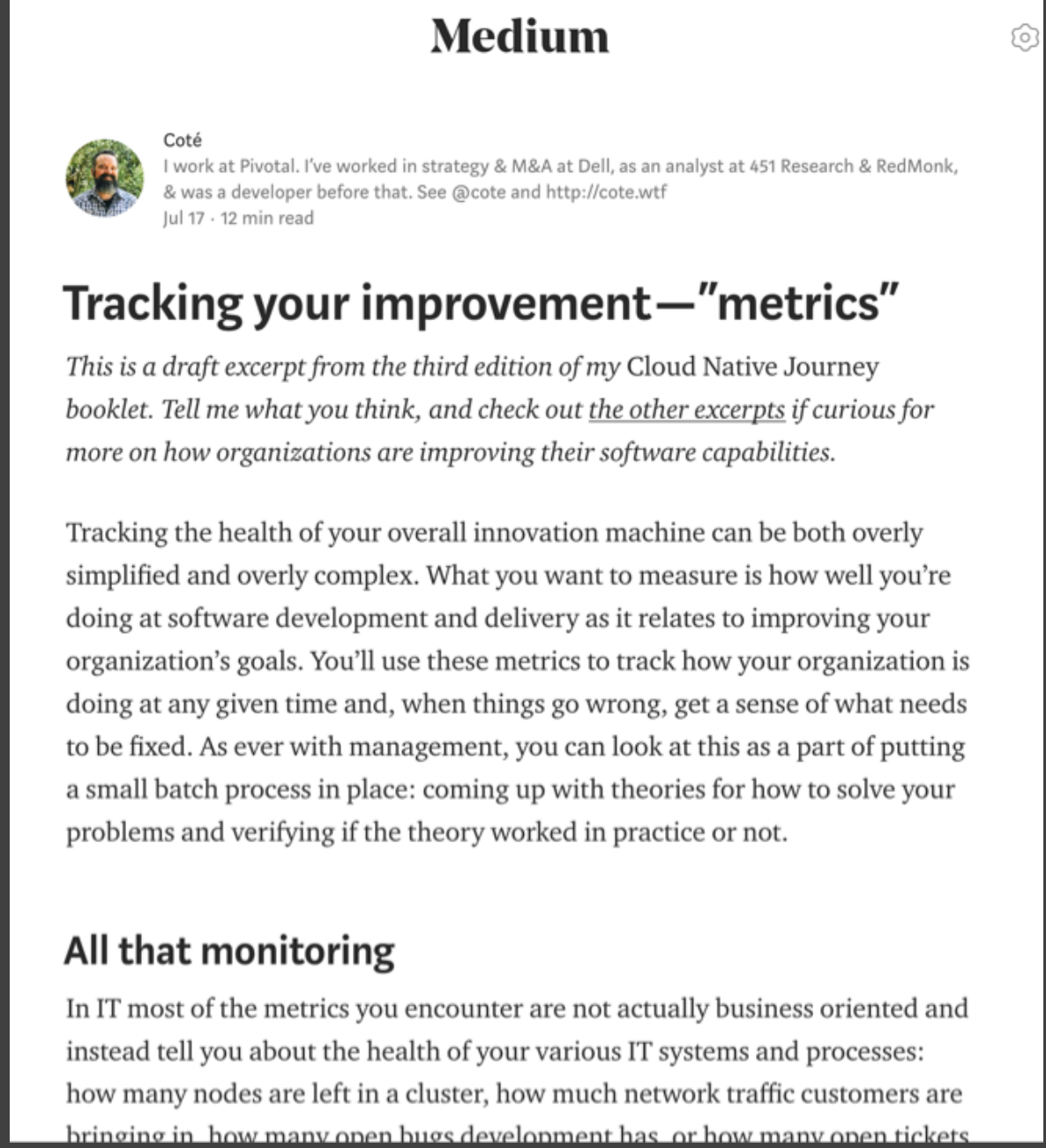
Spent

MVP yielded $214k/day in fuel savings
x 5 years to deploy tool w/ waterfall method (generous)
= **$391,000,000 Cost of Delay**

New methodology allowed the USAF to avoid $391M in opportunity cost—we can’t afford waterfall

Source: "Cost of Delay - How PCF Helped Demonstrate the DoD Can't Afford Business as Usual," Capt. Bryon Kroger & Tory Galvin, USAF, CF Summit NA 2018, April, 2018.

More


Medium

Coté
I work at Pivotal. I've worked in strategy & M&A at Dell, as an analyst at 451 Research & RedMonk, & was a developer before that. See @cote and http://cote.wtf
Jul 17 · 12 min read

Tracking your improvement—"metrics"

This is a draft excerpt from the third edition of my Cloud Native Journey *booklet. Tell me what you think, and check out the other excerpts if curious for more on how organizations are improving their software capabilities.*

Tracking the health of your overall innovation machine can be both overly simplified and overly complex. What you want to measure is how well you're doing at software development and delivery as it relates to improving your organization's goals. You'll use these metrics to track how your organization is doing at any given time and, when things go wrong, get a sense of what needs to be fixed. As ever with management, you can look at this as a part of putting a small batch process in place: coming up with theories for how to solve your problems and verifying if the theory worked in practice or not.

All that monitoring

In IT most of the metrics you encounter are not actually business oriented and instead tell you about the health of your various IT systems and processes: how many nodes are left in a cluster, how much network traffic customers are bringing in, how many open bugs development has, or how many open tickets

http://cote.coffee/bettersoftware/