



Pivotal.

The agile warfighter

Creating better software in defense

August, 2018

@cote

<http://cote.io/about>



Pivotal

 RedMonk

 451 Research[®]

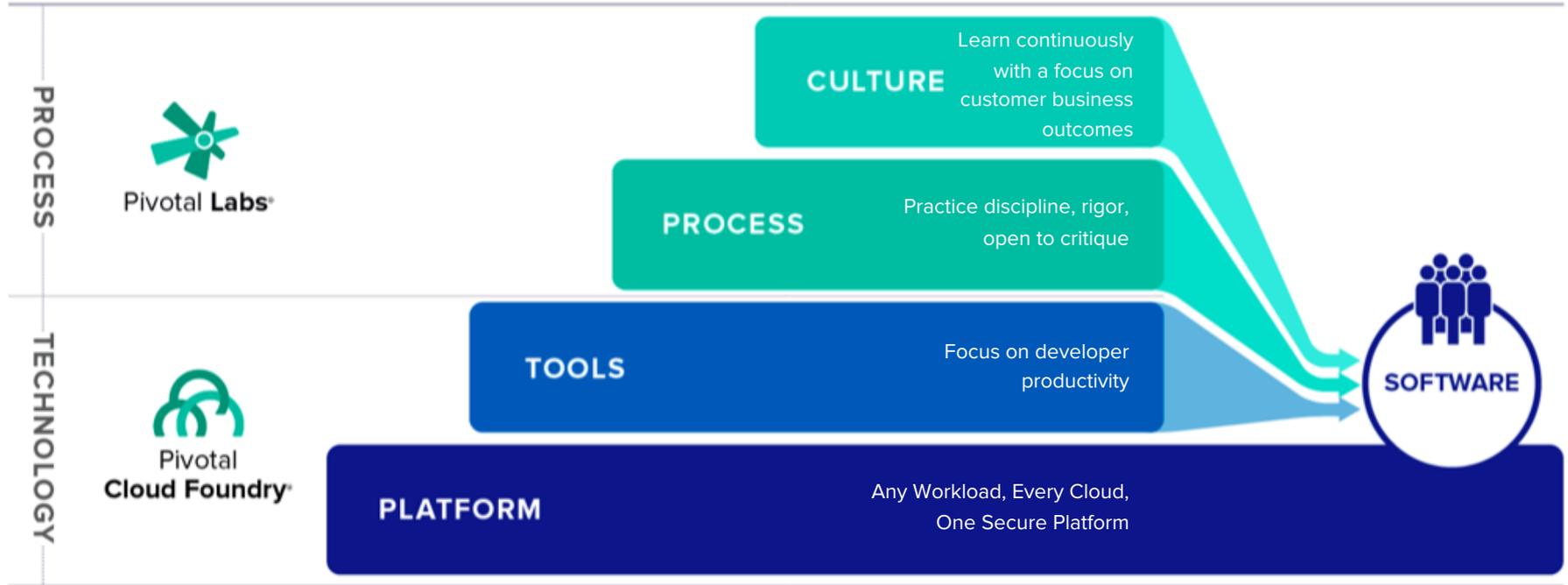
 SOFTWARE
DEFINED
TALK

 The Register[®]

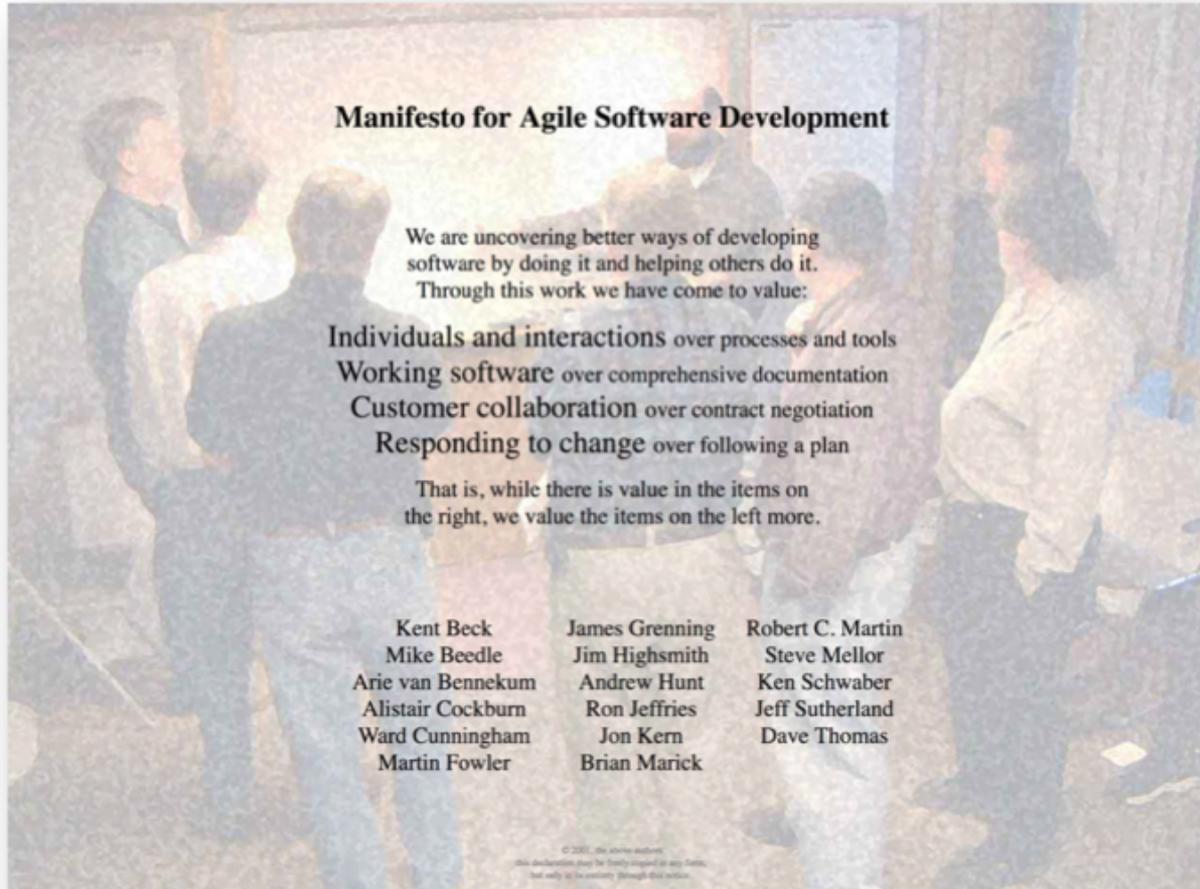
 DELL

 bmc software  Funds Xpress

Pivotal: transforming how the world builds software

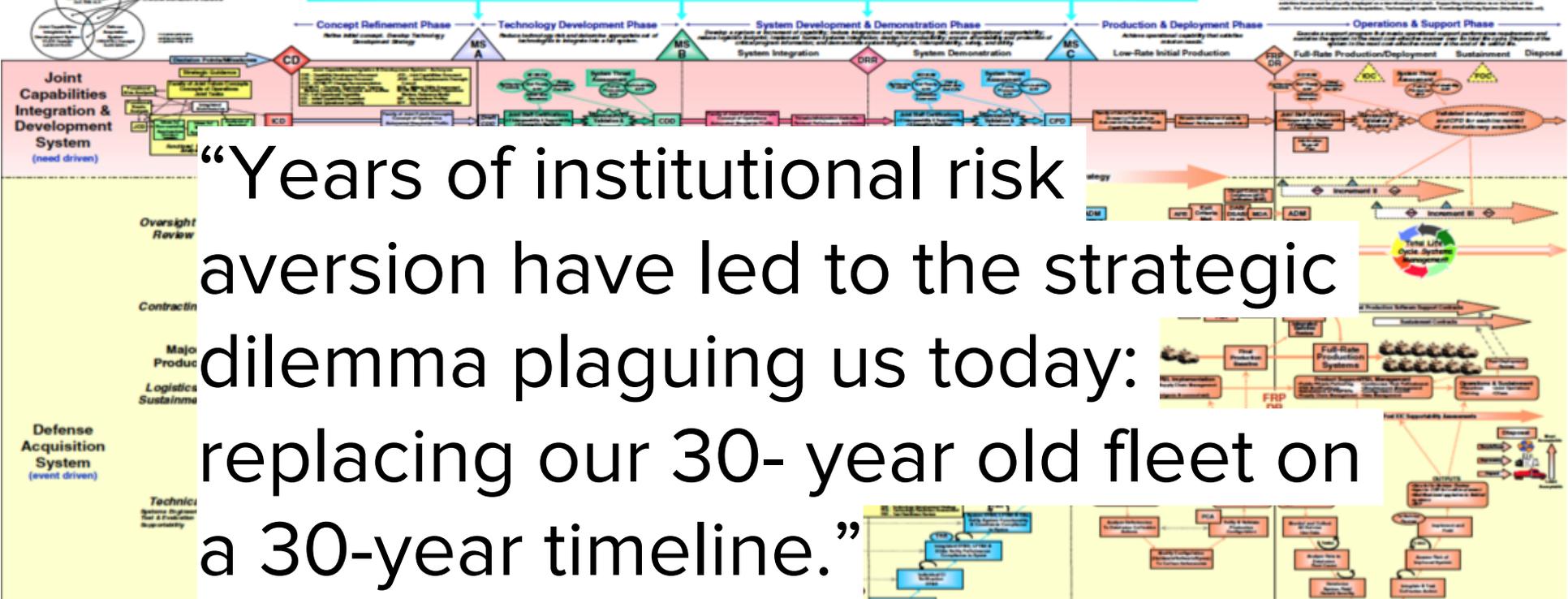


Agile warfighters need agile software



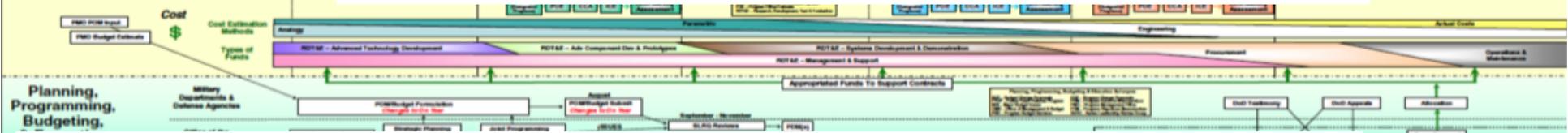
Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework

The Milestone Decision Authority may authorize entry into the acquisition process at any point, consistent with phase-specific entrance criteria and statutory requirements



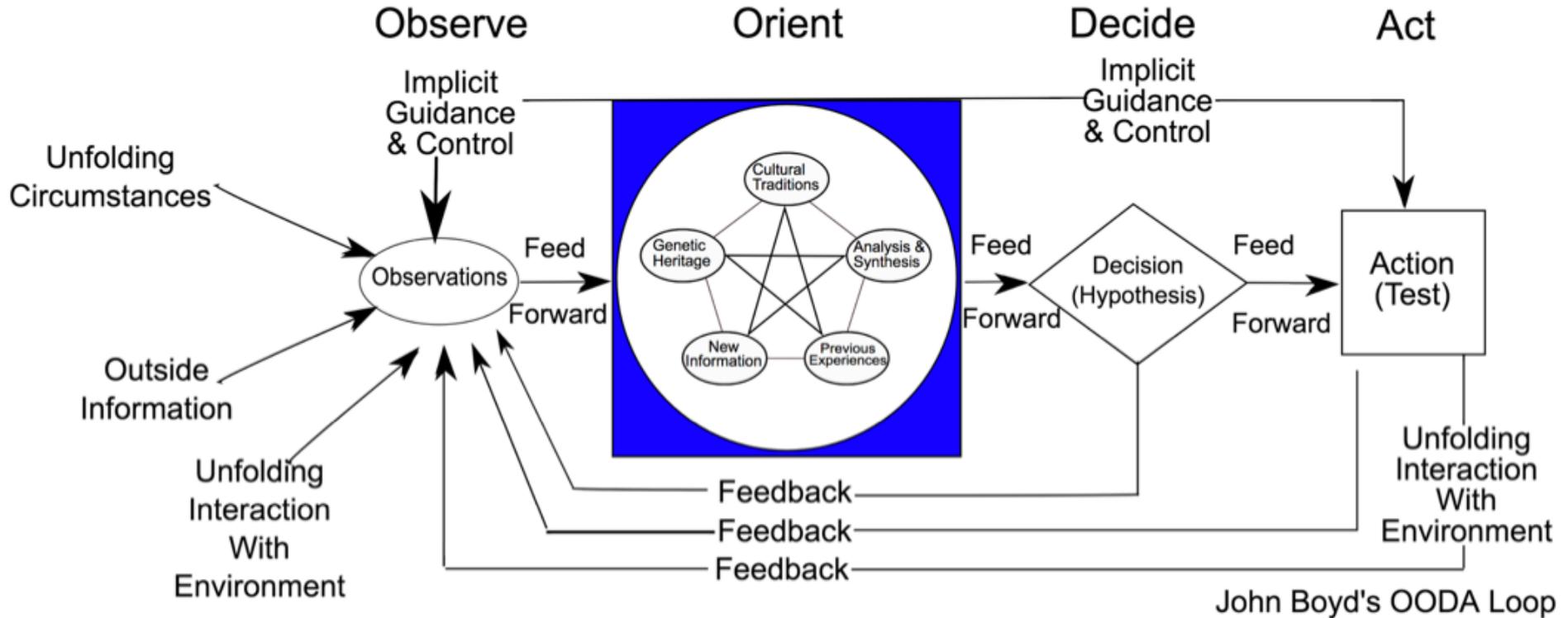
“Years of institutional risk aversion have led to the strategic dilemma plaguing us today: replacing our 30- year old fleet on a 30-year timeline.”

General James “Mike” Holmes, Commander, Air Combat Command



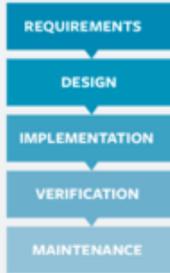
Source: ["Cost of Delay - How PCF Helped Demonstrate the DoD Can't Afford Business as Usual."](#) Capt. Bryon Kroger & Tory Galvin, United States Air Force, CF Summit NA 2018, April, 2018. ["From No Delivery to Continuous Delivery."](#) Adam Furtado, USAF, CF Summit 2018.

Surviving fast, agile combat

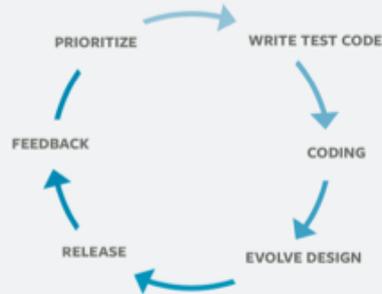


Fast feedback loops, leading to better design decisions

Traditional / Waterfall:
(every 6-12 months)



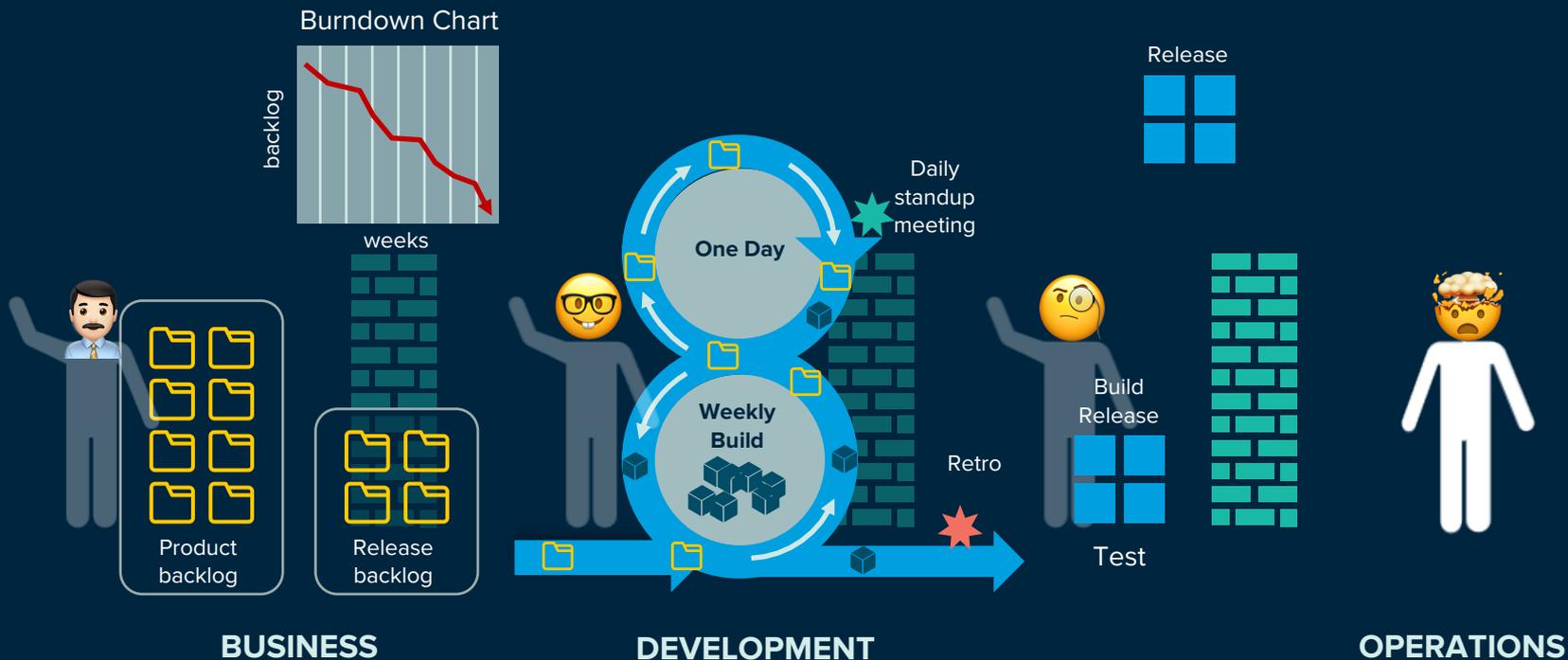
Iterative:
(daily/weekly)



Agile loop advantages:

- Real users drive requirements
- Better function fit & suitability
- Pay for only what's needed
- Delivers value instantly
- Learning the right solution
- Fully tested, faster recovery
- Legacy plague resistant
- Engaged staff

Modernizing developers creates a new bottleneck: ops

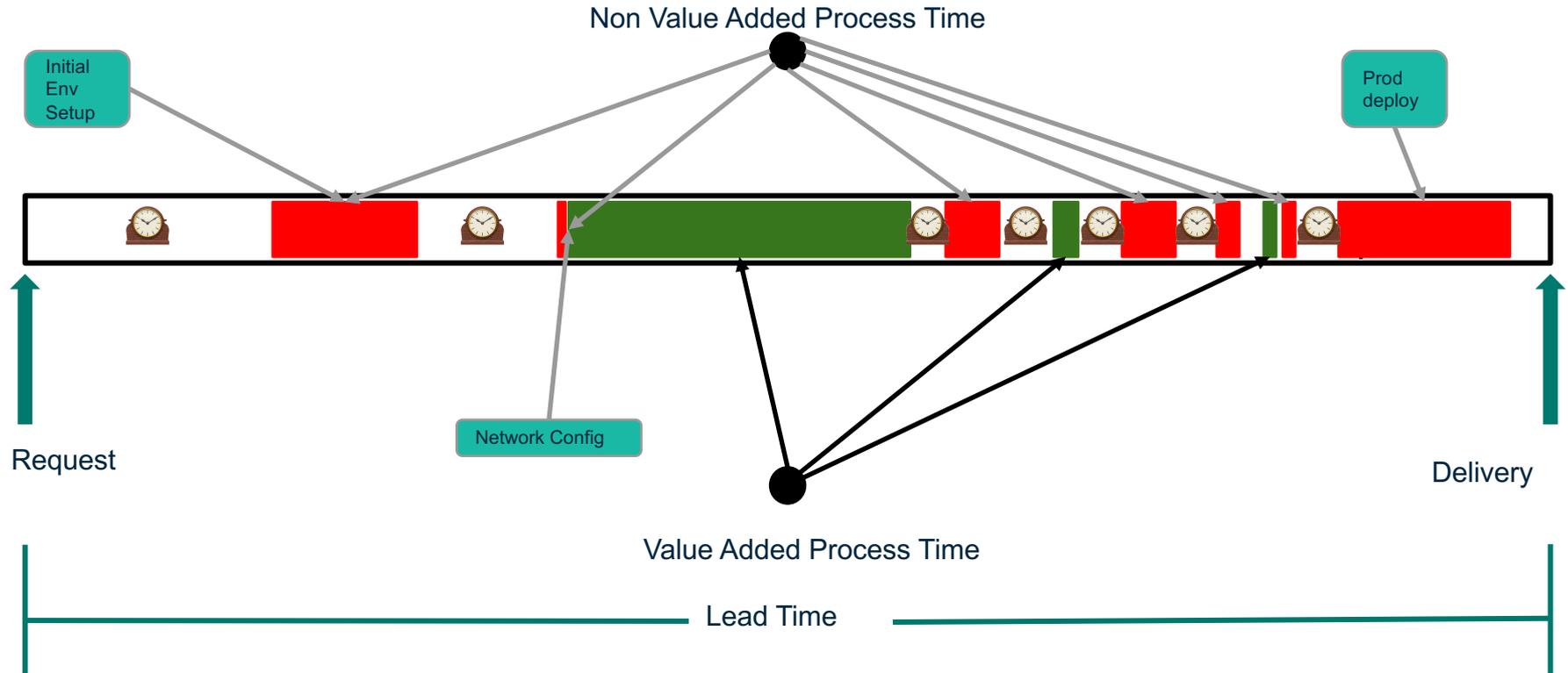




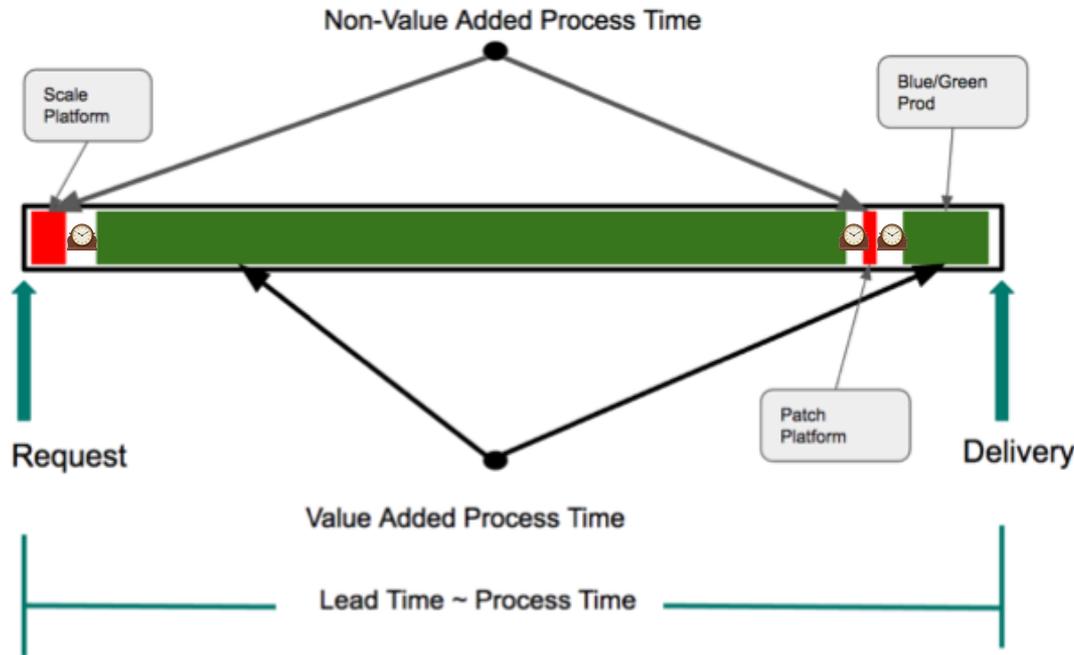
User

Typical value stream, 6 week lead time

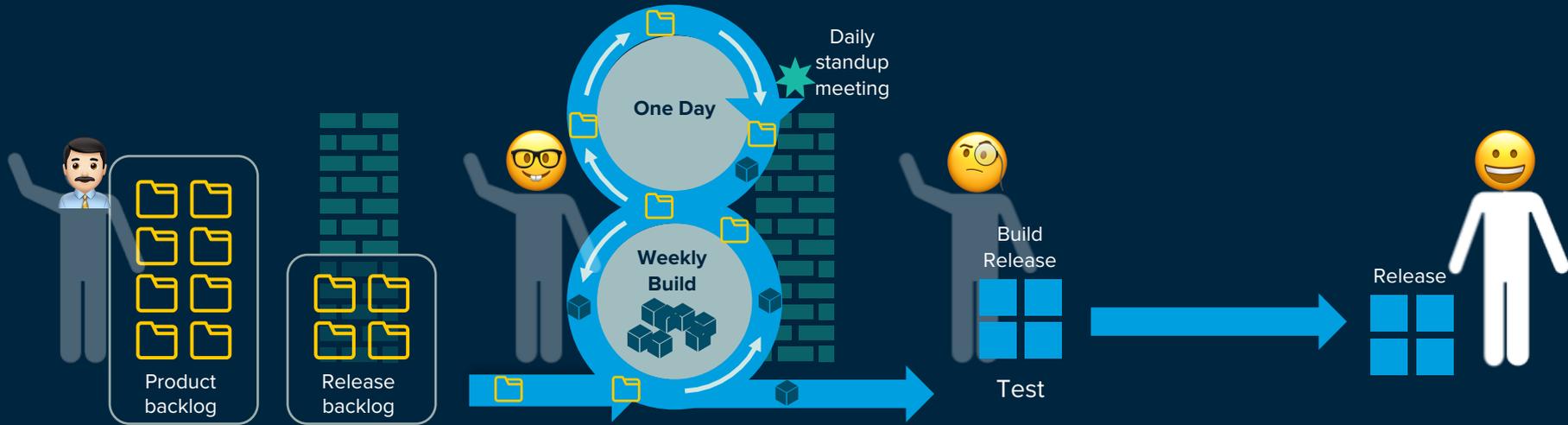
White = wait time
Red = ops time
Green = dev time



DevOps value stream, 3-6 weeks lead time



- Driven by ruthless automation in dev, testing, & ops
- PaaS automates most ops tasks & wait time
- ~85% reduction in ops time
- ~50% reduction in release timeline
- Option to halve release cycles for more feedback



BUSINESS

DEVELOPMENT

Platform as a Service



OPERATIONS

Private sector is agile...at least, the successful enterprises

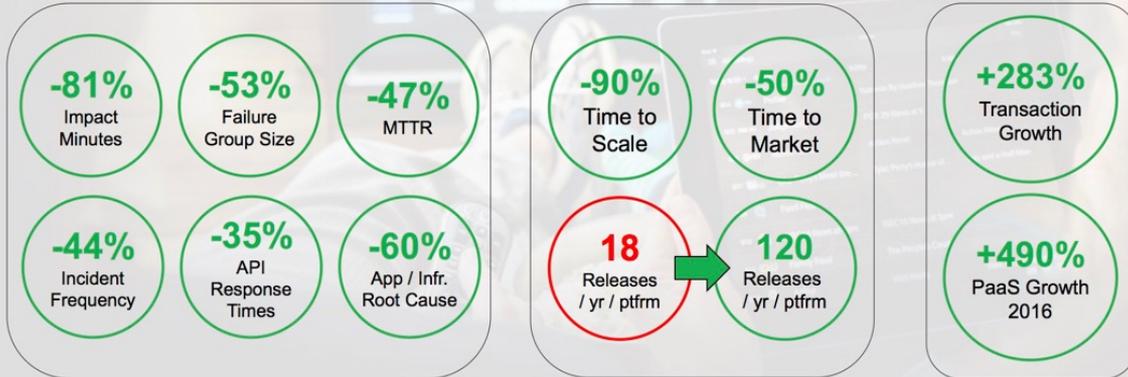
Our Results



Resiliency (run the business)

Time to Market (change the business)

Scale



- 5,700+ apps & services, customer-facing & back-office for Xfinity
- 25k transactions/second, 1.5bn/day
- 70% of BOSH deploys during the day
- 4 platform operators
- Compare to T-Mobile USA with 8 platform ops

Source: ["Comcast Cloud Foundry Journey - Part 2,"](#) Greg Otto, Comcast, CF Summit NA, June 2017; [Comcast SpringOne talk,](#) Dec 2017; ["Comcast Cloud Foundry Journey,"](#) Greg Otto, Comcast, June 2013; ["Zero to 12 Million,"](#) Brendan Aye, T-Mobile USA, SpringOne Platform, Dec 2017 (11,000 containers in prod. In Dec 2017: 6 ops, 2 dev).



Source: Combined Air Operations Command (CAOC), Qatar, U.S. Air Force photo by Staff Sgt. Alexander W. Riedel.

38SLD 22941 53104



▲ KESSEL RUN HEADQUARTERS

TER # 235235 MGRS 34UCE 41427 49012
Nearest City Mos Eisley Province Tatooine Country SW
Last Struck - Strike Status **Approved** Killbox 93AE

LOA'S WITHIN 5KM Include Archived Targets

Cantina

38SLD 54423 44342
2KM NW OF LOA
TER - 12678
Last Active TODAY
Last Struck 14 JUL 17
Status - Current

Enemy Lightsaber Factory

38SLD 54423 44342
3KM NE OF LOA
TER - 12680
Last Active 14 JUL 17
Last Struck 11 JUL 17
Status - Struck

Skywalker HVI

38SLD 54423 44342
3KM S OF LOA
TER - 1230985
Last Active TODAY
Last Struck 13 JUN 17
Status - Working

Jedi Staging Area

38SLD 54423 44342
3KM NW OF LOA
TER - 12678
Last Active TODAY
Last Struck 14 JUL 17
Status - Current

Millenium Falcon Facility

38SLD 54423 44342
3KM SW OF LOA
Status - Current

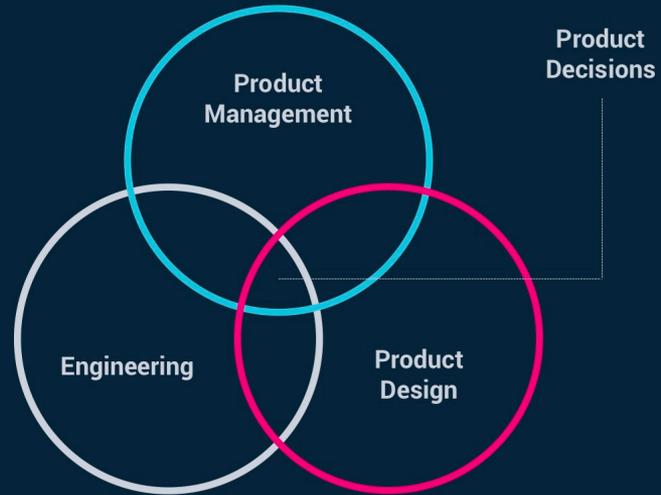




Source: [Lieutenant Colonel Enrique Oti keynote](#), SpringOne Platform 2017, Dec 2017.



balancedteams



Agile



Platform as a Service



8 hours → <2 hours

6 operators → 1 operator

5 years → 120 days → weekly

\$2.2m project

~\$214k/day fuel savings

USAF Follow-on: ~16 apps in production by 2019



Oct. 2016 - Whiteboard



April 2017 - JIGSAW



Dec. 2017 - CHAINSAW



March 2018 - RAVEN



May 2018 – 5 prod. apps, 8 in dev



Jan 2019 – plan for 15-18 prod. apps

- JIGSAW: \$2.2m
- Avoided \$391m cost of delay
- 1 new feature a week, at least
- 124 days to production, avg.
- Contacts awarded in 60 days
- Proven method spreading to other groups
- Canceled existing \$745m contract

Note: apps in time is a sampling of public comments. Sources: ["Cost of Delay - How PCF Helped Demonstrate the DoD Can't Afford Business as Usual,"](#) Capt. Bryon Kroger & Tory Galvin, USAF, April, 2018; ["The U.S. Air Force learned to code - and saved the Pentagon millions,"](#) Mark Wallace, Fast Company, July 5th, 2018; [Lieutenant Colonel Enrique Oti keynote](#), SpringOne Platform 2017, Dec 2017.

Agile Software → Agile Warfighters

These transformations are real, across industries

AUTO & TRANSPORTATION



INDUSTRIAL & BUSINESS SVCS.



FINANCIAL SERVICES



HEALTHCARE & INSURANCE



TECHNOLOGY & MEDIA



CONSUMER & COMMUNICATIONS



“The warfighter can’t wait until it’s perfect.”



“A [waterfall] mistake could cost \$100 million, likely ending the career of anyone associated with that decision. A smaller mistake is less often a career-ender and thus encourages smart and informed risk-taking.”



“With a more agile approach, we pick a place to start and get to a point where you can have an intelligent conversation... a point where the requirements are 80% done and the application is good enough.

[M. Wes Haga, US Air Force](#)

Sources: [“How the US Air Force Made Its ISR Network Cheaper to Run and Easier to Upgrade.”](#) M. Wes Haga, Oct, 2017; [“Air Force Intelligence Unit Goes Agile.”](#) Charles Babcock, *Information Week*, June, 2017; [“Limit upfront analysis by including frequent, real-world feedback from users.”](#) Coté, Nov 2017; [sticky-staring team from USAF Kessel Run Group](#); title quote from Capt. Bryon Kroger, Kessel Run team.

From coding 20% of the time coding to coding 90% of the time

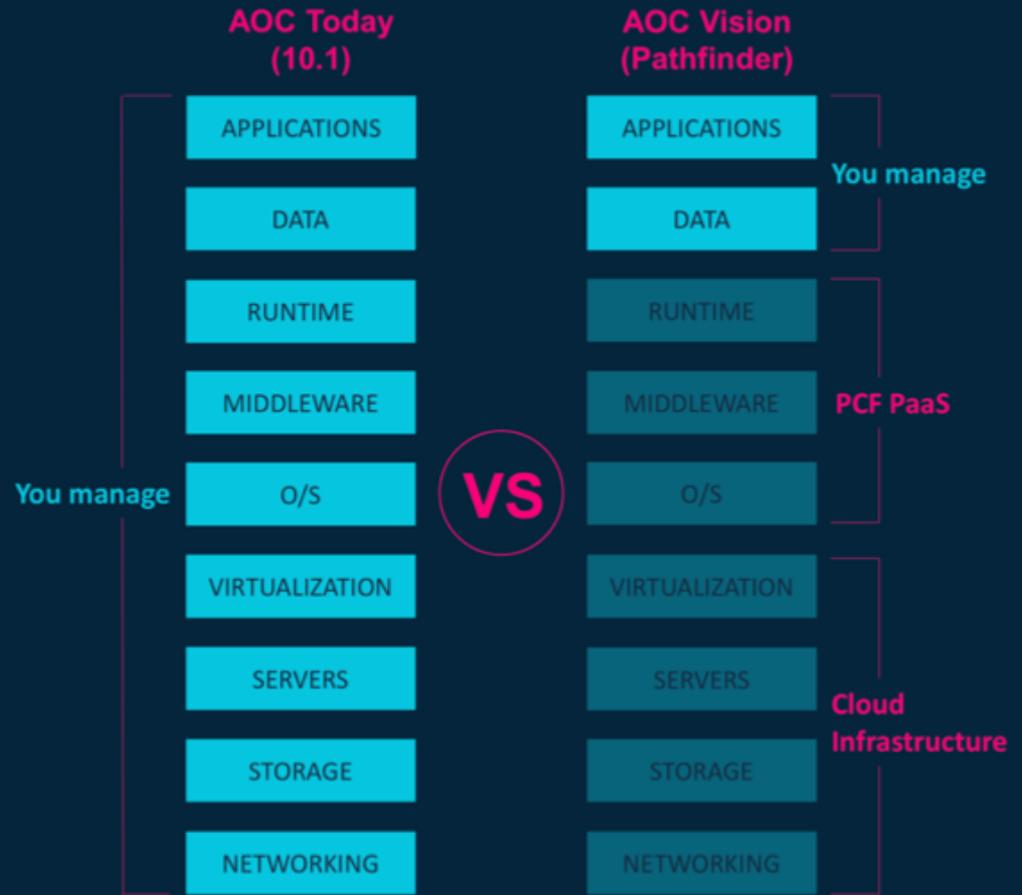


An agile methodology, proven over 25+ years:

- Balanced teams w/all roles needed, dedicated to the product
- Paired programming, & beyond
- Test-driven Development (TDD)
- Short iterations
- Continuous Integration & Continuous Delivery

automategrc

- 80-90% of Risk Management Framework **controls inherited** via IaaS/PaaS
- Cut **Authority-to-Operate process** from ~10 months to <1 week for ~40 remaining controls at application layer



Source: "Cost of Delay - How PCF Helped Demonstrate the DoD Can't Afford Business as Usual," Capt. Bryon Kroger & Tory Galvin, United States Air Force, April, 2018. ["Zero to 12 Million."](#) Brendan Aye, T-Mobile USA, SpringOne Platform, Dec 2017.

Defense Innovation Board Metrics for Software Development

Version 0.9, last modified 9 Jul 2018

Software is increasingly critical to the defense is plagued by poor quality and that software complexity is often estimated rate of progress is measured in terms of are easily measured, they are not needs are especially suspect as measurements needed functionality and value to users within DoD programs using these observations misleading. As an alternative, we believe performance for software programs and

#	Metric
1	Time from program launch to deploy simplest useful functionality
2	Time to field high priority fix (spec → fix newly found security hole (find →
3	Time from code committed to code in
4	Time req'd for full regression test (aud and cybersecurity audit/penetration t
5	Time required to restore service after
6	Automated test coverage of specs / c
7	Number of bugs caught in testing vs i
8	Change failure rate (rollback deploys)
9	% code available to DoD for inspection/rebuild
10	Complexity metrics
11	Development plan/environment metri
12	"Nunn-McCurdy" threshold (for any r

Defense Innovation Board Ten Commandments of Software

Version 0.14, last modified 15 April 2018
Department of Defense
OFFICE OF PUBLICATION AND SECURITY REVIEW

CLEARED
For Open Publication
Apr 20, 2018 5

Executive Summary

The Department of Defense (DoD) must be able to develop and deploy software as fast or faster than its adversaries are able to change tactics, building on commercially available tools and technologies. Recognizing that "software" can range from off-the-shelf, non-customized software to highly-specialized, embedded software running on custom hardware, it is critical that the right tools and methods be applied for each type. In this context we offer the following ten "commandments" of software acquisition for the DoD:

1. Make computing, storage, and bandwidth abundant to DoD developers and users.
2. All software procurement programs should start small, be iterative, and build on success – or be terminated quickly.
3. Budgets should be constructed to support the full, iterative life-cycle of the software being procured with amount proportional to the criticality and utility of the software.
4. Adopt a DevOps culture for software systems.
5. Automate testing of software to enable critical updates to be deployed in days to weeks, not months or years.
6. Every purpose-built DoD software system should include source code as a deliverable.
7. Every DoD system that includes software should have a local team of DoD software experts who are capable of modifying or extending the software through source code or API access.
8. Only run operating systems that are receiving (and utilizing) regular security updates for newly discovered security vulnerabilities.
9. Data should always be encrypted unless it is part of an active computation.
10. All data generated by DoD systems - in development and deployment - should be stored, mined, and made available for machine learning.

Vendors must also respond to evolving "threats"

- Minimal Suitable Products - smaller projects that show value quickly.
- Adapting requirements & scope - smaller, agile contracts that can change.
- Staff that can enable, model, and augment agile product development.
- Adapting & managing cloud native platforms for secure, private cloud use.
- DIUx model - partner with smaller, new firms.
- A willingness to change, demonstrated by leadership and budget.



BOOK 2

PART 1:

- 1
- 2
- 3
- 4
- 5

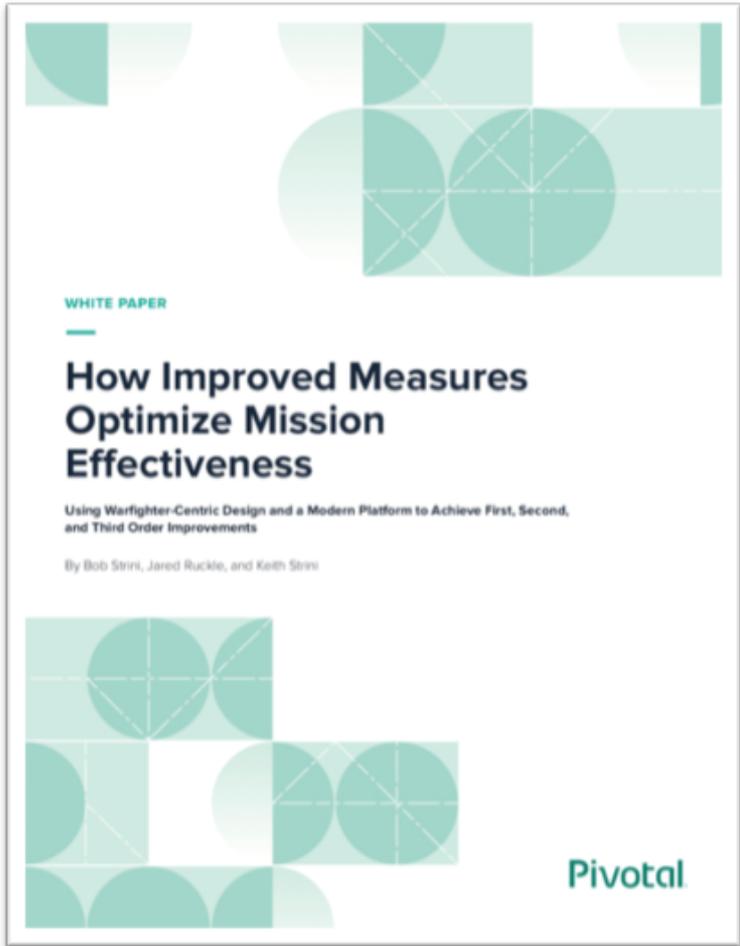
PART 2:

- Cloud-Native Transformation—Doing the Work 23
 - Creating the Game 23
 - Creating a Continuous Learning Organization 25
 - Building Trust and Deformalizing Stale Corporate Culture 25
 - The Three Finance Questions You Meet in Data Conference Rooms 28
 - Success is the Best ROI 30
 - Case Study: IRS 30
 - Case Study: "It Was Why Beyond What We Needed to Even be Doing" 33
 - Risk Management with Small Batches 33
 - Security: Same Old Story, New Sully Tools 34
 - Revitalizing Legacy Code 38
 - Living with Legacy 39
 - Avoiding Legacy Pitfalls 40
 - Case Study: Avoiding Portfolio Paralysis Analysis 42
- Cloud Platform Reference Architecture 43
- Building Your Own Cloud Platform is Probably a Bad Idea 44
- For Just \$1M, You Too, Can Have Your Very Own Platform in Two Years 45
- Pivotal Cloud Foundry 45
- Picking Projects by Portfolio Pondering 47
- Planning Out the Initial Project 48
- Crafting Your Hacky Stick 49

Changing Your Process and Organization 6

- Smart Batch Thinking 7
- Become a Learning, not a Status-Checking Organization 8
- Moving from future management to risk management 8
- Case Study: "What Have You Learned?" 10
- Shift to User-Centric Design 10
- Case Study: No One Wants to Call the IRS 11
- How much do I owe the IRS? 11
- DevOps as the Overarching Process 12
- Put it All Together to Enable Small Batches 14
- The Shape of a Cloud-Native Organization 14
- Product-Balanced Teams 15

Pivotal



“We are uncovering better ways of developing software by doing it and helping others do it.”

- [The Agile Manifesto](#), 2001

Thanks!

@cote | cote@pivotal.io