improving
It's what we do. ™

The nitty-gritty practices of agile coaching

What is the goal of Agile?

Move Quicker?

Get more stuff done?

Save Money?

User Feedback! We want lots of it!

How often is your Scrum team releasing to production?

Everyday?

Weekly?

Monthly?

Yearly?

I don’t know hopefully soon?

Allison Pollard

As an agile coach with Improving in Dallas, Allison Pollard helps people discover their agile instincts and develop their coaching abilities. Allison is also a Certified Professional Co-Active Coach, a foodie, and proud glasses wearer.

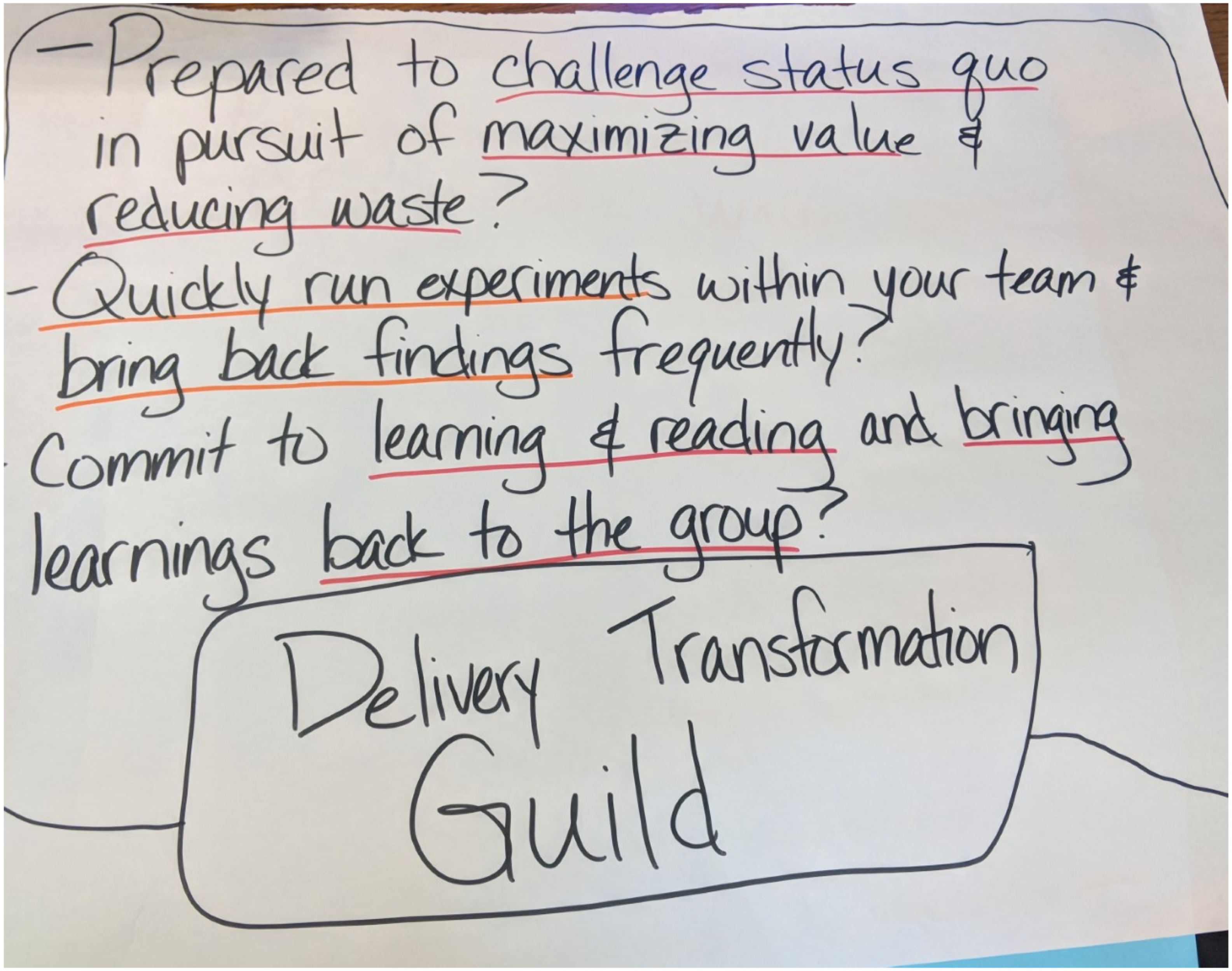
Jason Hobbs

Jason Hobbs is an Agile-minded leader at American Airlines whose mission, having himself survived many terrible environments as a software developer, is to create a better world for technologists and software development teams, and as a result allow the world to enjoy better software and applications.

Companies adopt Agile to get better results—not for the sake of adopting Agile.

How can I evaluate if my coaching is achieving better results?

Generative Culture

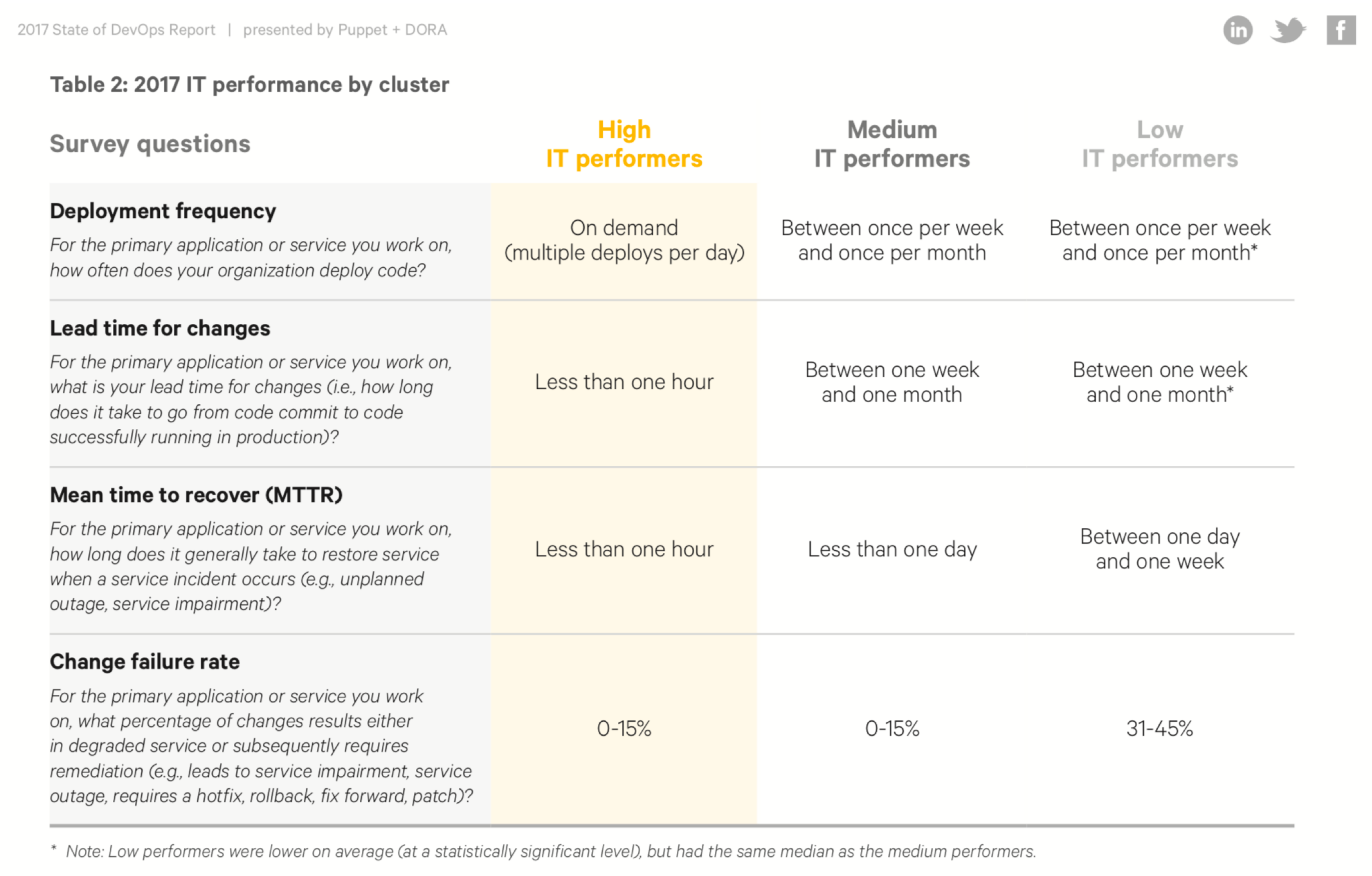


- Prepared to challenge status quo in pursuit of maximizing value & reducing waste?
- Quickly run experiments within your team & bring back findings frequently?
- Commit to learning & reading and bringing learnings back to the group?

Delivery Transformation Guild

IT performance *predicts* organizational performance

2017 State of DevOps Report



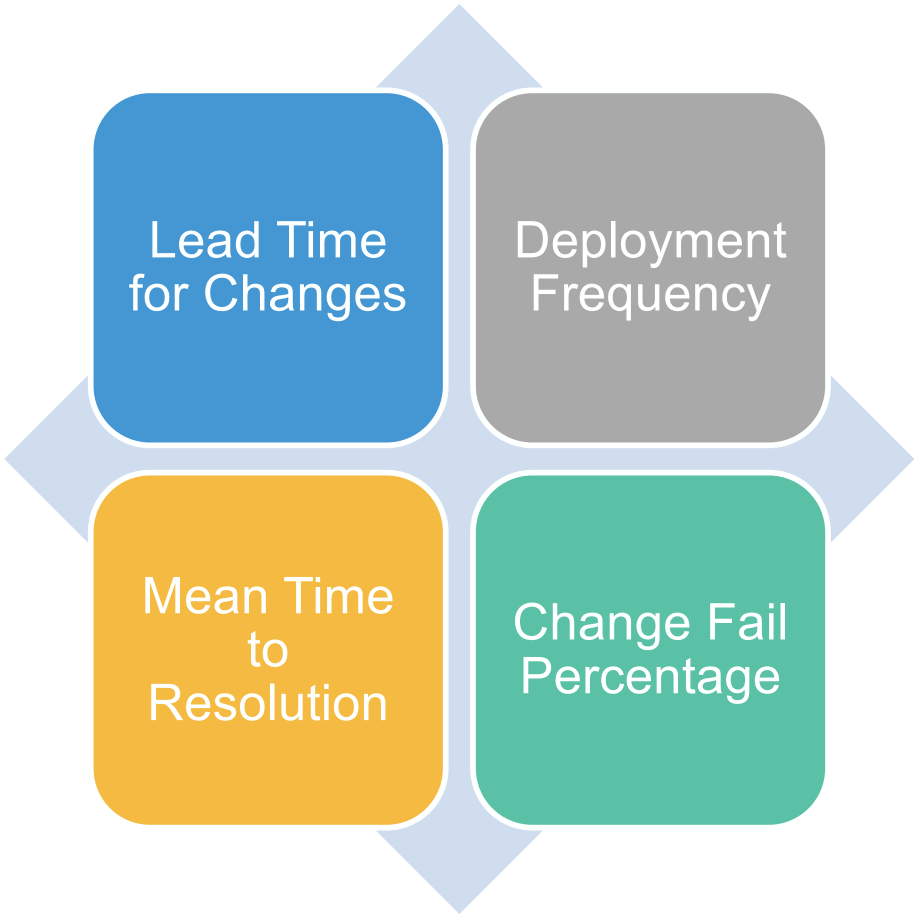
2017 State of DevOps Report | presented by Puppet + DORA

Table 2: 2017 IT performance by cluster

Survey questions	High IT performers	Medium IT performers	Low IT performers
Deployment frequency *For the primary application or service you work on, how often does your organization deploy code?*	On demand (multiple deploys per day)	Between once per week and once per month	Between once per week and once per month*
Lead time for changes *For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code commit to code successfully running in production)?*	Less than one hour	Between one week and one month	Between one week and one month*
Mean time to recover (MTTR) *For the primary application or service you work on, how long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?*	Less than one hour	Less than one day	Between one day and one week
Change failure rate *For the primary application or service you work on, what percentage of changes results either in degraded service or subsequently requires remediation (e.g., leads to service impairment, service outage, requires a hotfix, rollback, fix forward, patch)?*	0-15%	0-15%	31-45%

* *Note: Low performers were lower on average (at a statistically significant level), but had the same median as the medium performers.*

The team scoreboard and the metrics of change



Discuss:

- How does strong performance in this delivery capability help organizations?
- Imagine your team is a high performer in this metric—you are deploying to production multiple times a day. What does it feel like?
- What changes or practices could a team experiment with to improve this metric?

Lead Time for Changes

How long it takes to go from code commit to code successfully running in production for the primary service or application they work on

From *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* by Gene Kim, Jez Humble, and Nicole Forsgren

Deployment Frequency

How often their organization deploys code for the primary service or application they work on

Mean Time to Resolution

How long it generally takes to restore service for the primary application or service they work on when a service incident (e.g., unplanned outage, service impairment) occurs

From *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* by Gene Kim, Jez Humble, and Nicole Forsgren

Change Fail Percentage

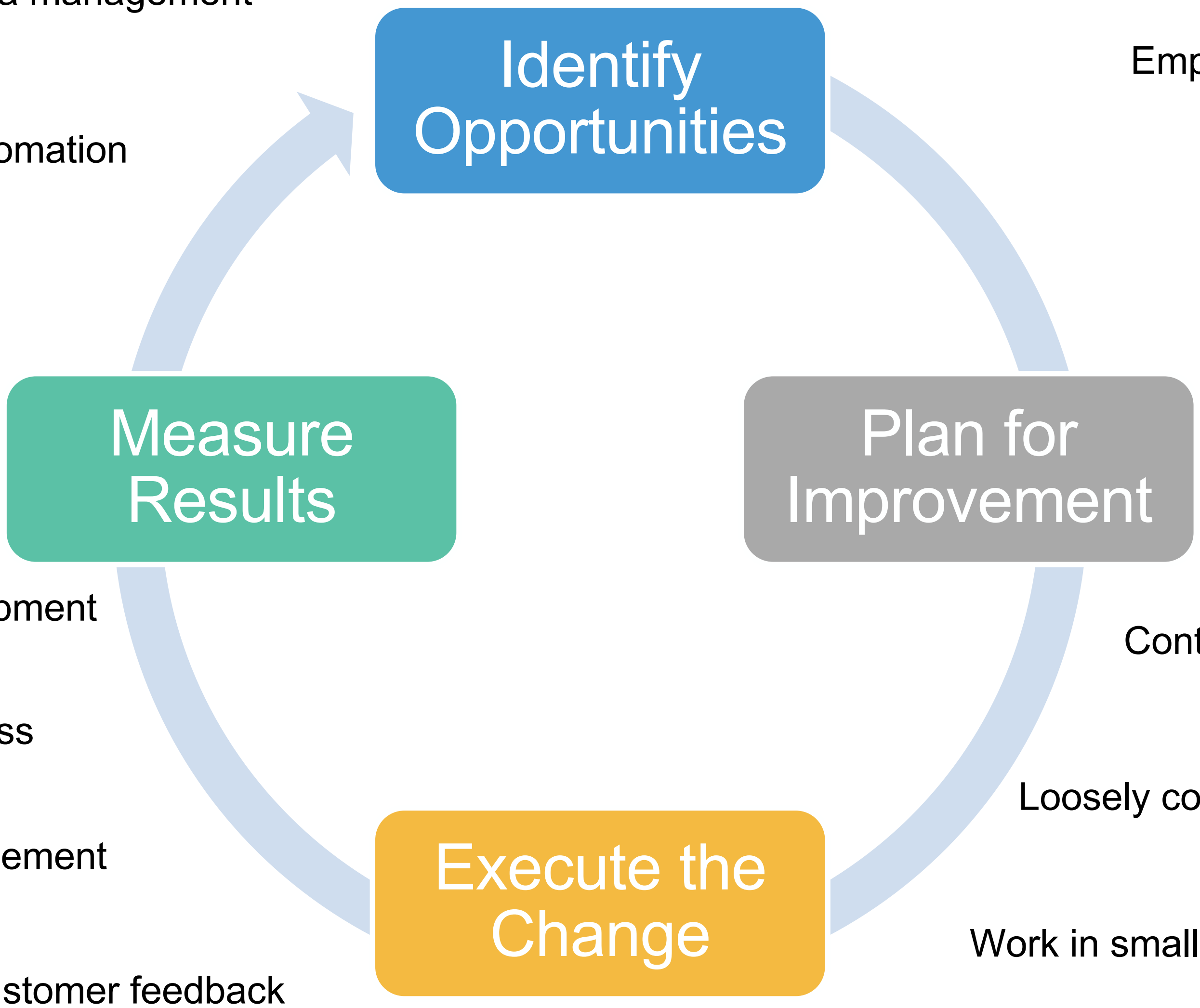
Percentage of changes for the primary application or service they work on either result in degraded service or subsequently require remediation

From *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations* by Gene Kim, Jez Humble, and Nicole Forsgren

improving
It's what we do.™

Continuous Improvement

How can we get better?



Version control

Make flow of work visible

Empowered teams

Test automation

Continuous integration

Loosely coupled architecture

Work in small batches

Gather customer feedback

Visual management

Limit work in process

Trunk-based development

Shift Left on Security

Deployment automation

Test data management

From theory...to application

What’s your Next Step?

Contact Us

Allison Pollard
@allison_pollard
http://www.allisonpollard.com/

Jason Hobbs
@telnetdoogie

Deployment frequency

For the primary application or service you work on, how often does your organization deploy code?

In 2017, research shows high performers have 46 times more frequent code deployments than their lower- performing peers. How does strong performance in this delivery capability help organizations?

Imagine your team is a high performer in this metric—you are deploying to production multiple times a day. What does it feel like?

What changes or practices could a team experiment with to improve this metric?

Notes

Lead time for changes

For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code commit to code successfully running in production)?

In 2017, research shows high performers have 440 times faster lead time from commit to deploy than their lower- performing peers. How does strong performance in this delivery capability help organizations?

Imagine your team is a high performer in this metric—you are deploying to production multiple times a day. What does it feel like?

What changes or practices could a team experiment with to improve this metric?

Notes

In 2017, research shows high performers have 96 times faster mean time to recover from downtime than their lower- performing peers. How does strong performance in this delivery capability help organizations?

Imagine your team is a high performer in this metric—you are deploying to production multiple times a day. What does it feel like?

Mean time to restore (MTTR)
For the primary application or service you work on, how long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?

What changes or practices could a team experiment with to improve this metric?

Notes

In 2017, research shows high performers have 5 times lower change failure rate (changes are 1/5 as likely to fail) than their lower- performing peers. How does strong performance in this delivery capability help organizations?

Imagine your team is a high performer in this metric—you are deploying to production multiple times a day. What does it feel like?

Change failure rate

For the primary application or service you work on, what percentage of changes results either in degraded service or subsequently requires remediation (e.g., leads to service impairment, service outage, requires a hotfix, rollback, fix forward, patch)?

What changes or practices could a team experiment with to improve this metric?

Notes

improving