ADDO
ALL DAY DEVOPS

OCTOBER 28, 2021

Jeremy Meiss

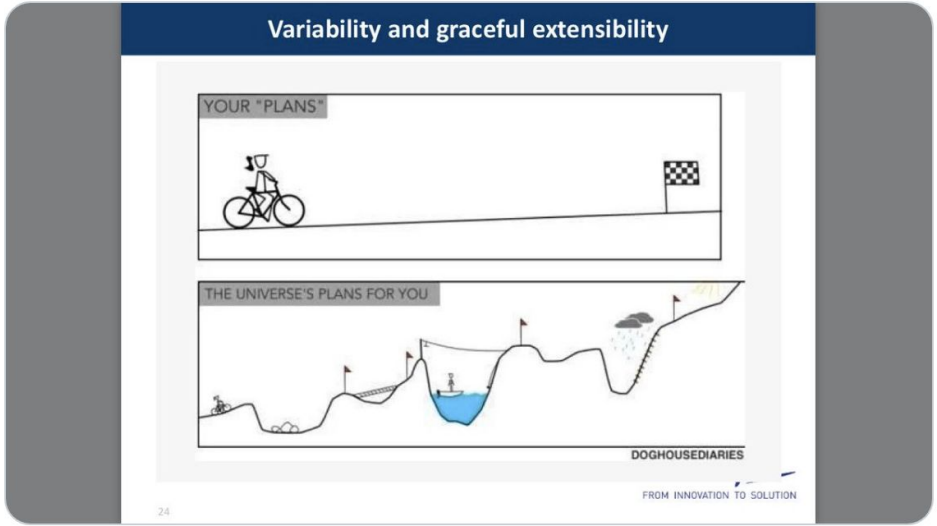
What a global pandemic can tell you about better DevOps practices


2016
State of DevOps Report
2017
State of DevOps Report
Presented by:
puppet
DORA
Sponsored by:
splunk>
amazon web services
Atlassian
Hewlett Packard Enterprise
WAVEFRONT by vmware
Electric Cloud
Deloitte.
2018
State of DevOps Report
puppet
splunk>
aws
Cognizant
2019
State of DevOps Report
puppet
circleci
splunk>
ANITIAN
servicenow

@IAmJerdog

John Allspaw
@allspaw
Normal 0%

Work-as-imagined versus work-as-done



3:00 AM · Apr 28, 2016 · Twitter for iPhone

@IAmJerdog

performance described
vs
performance derived

@IAmJerdog

Jeremy Meiss

Director, DevRel & Community

circleci

ADDO
ALL DAY DEVOPS

@IAmJerdog

2 million
jobs/day

44,000+
orgs

** 40k in 2019*

160,000+
projects

** 150k in 2019*

1,000x
Larger than surveys

@IAmJerdog

Four classic metrics

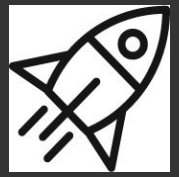
Deployment frequency

Lead time to change

Change failure rate

Recovery from failure time

CI/CD Benchmarks for high performance



Throughput

At will

Duration

<10 minutes

Success Rate

> 90%

Mean Time to Recovery

<1 hour

@IAmJerdog

The Data

Photo by: Matthew Henry

Throughput

Percentile	2020 Value	2019 Value
5p	0.03	0.03
50p	0.70	0.80
90p	16.03	13.00
95p	32.125	25.47
Mean	8.22	5.76

@IAmJerdog

Most teams are not deploying dozens of times per day

@IAmJerdog

Image by Pawan Kolhe from Pixabay

Duration

Percentile	2020 Value	2019 Value
5p	12 sec	10 sec
50p	3.96 min	3.38 min
90p	21.35 min	19.18 min
95p	34.01 min	31.73 min
Mean	24.6 min	26.76 min

@IAmJerdog

Photo by Lukas from Pexels

Success Rate

Percentile	2020 Value	2019 Value
5p	0%	0%
50p	61%	60%
90p	100%	100%
95p	100%	100%
Mean	54%	54%

@IAmJerdog

Photo by Brett Sayles from Pexels

Recovery Time

Percentile	2020 Value	2019 Value
5p	2.06 min	2.83 min
50p	55.11 min	52.5 min
90p	39 hours	47 hours
95p	3.4 days	3.93 days
Mean	14.85 hours	16.61 hours

@IAmJerdog

Recovery Time

Percentile	2020 Value	2019 Value
5p	2.06 min	2.83 min
50p	55.11 min	52.5 min
90p	39 hours	47 hours
95p	3.4 days	3.93 days
Mean	14.85 hours	16.61 hours

@IAmJerdog

Recovery Time

Percentile	2020 Value	2019 Value
5p	2.06 min	2.83 min
50p	55.11 min	52.5 min
90p	39 hours	47 hours
95p	3.4 days	3.93 days
Mean	14.85 hours	16.61 hours

@IAmJerdog

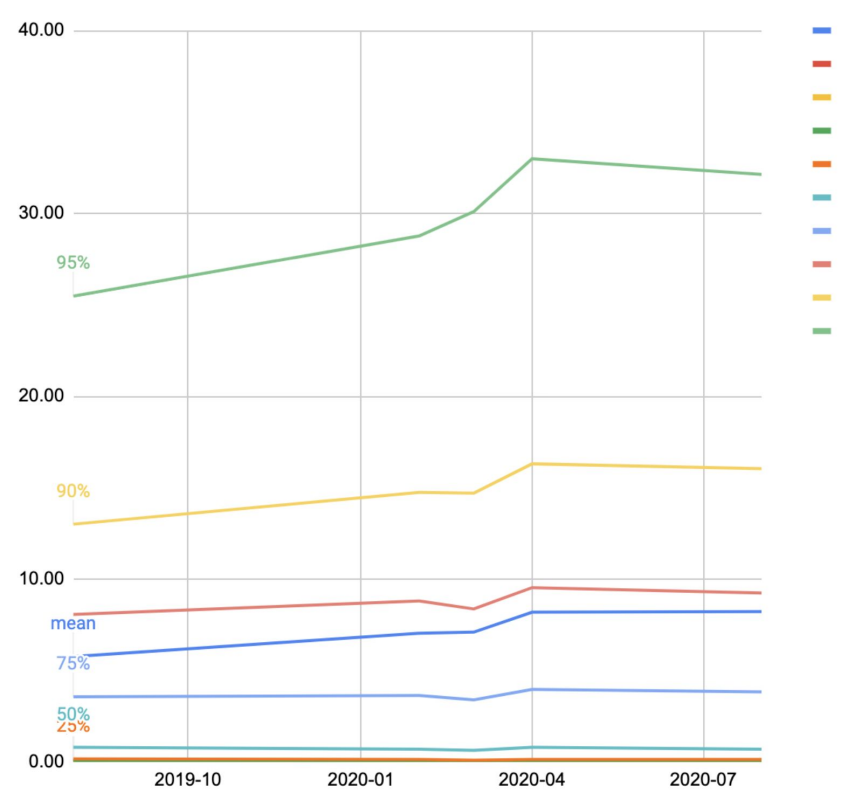
The Insight

2020 has been
a year.

@IAmJerdog

Throughput

Throughput in a global pandemic



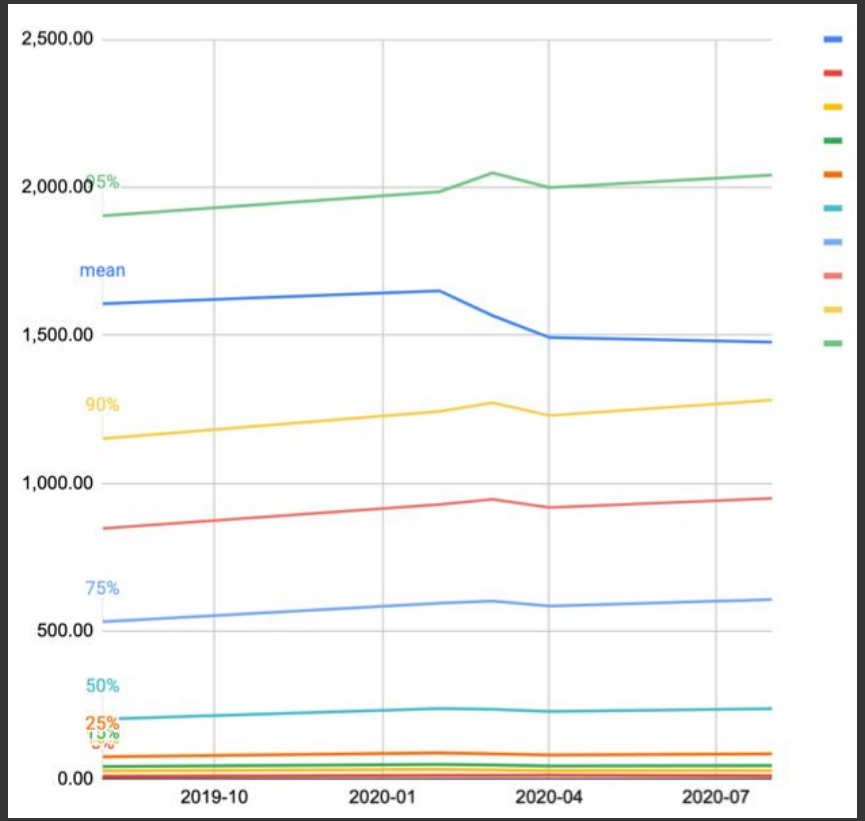
@IAmJerdog

Peak Throughput was in April 2020

@IAmJerdog

Duration

Duration in a global pandemic

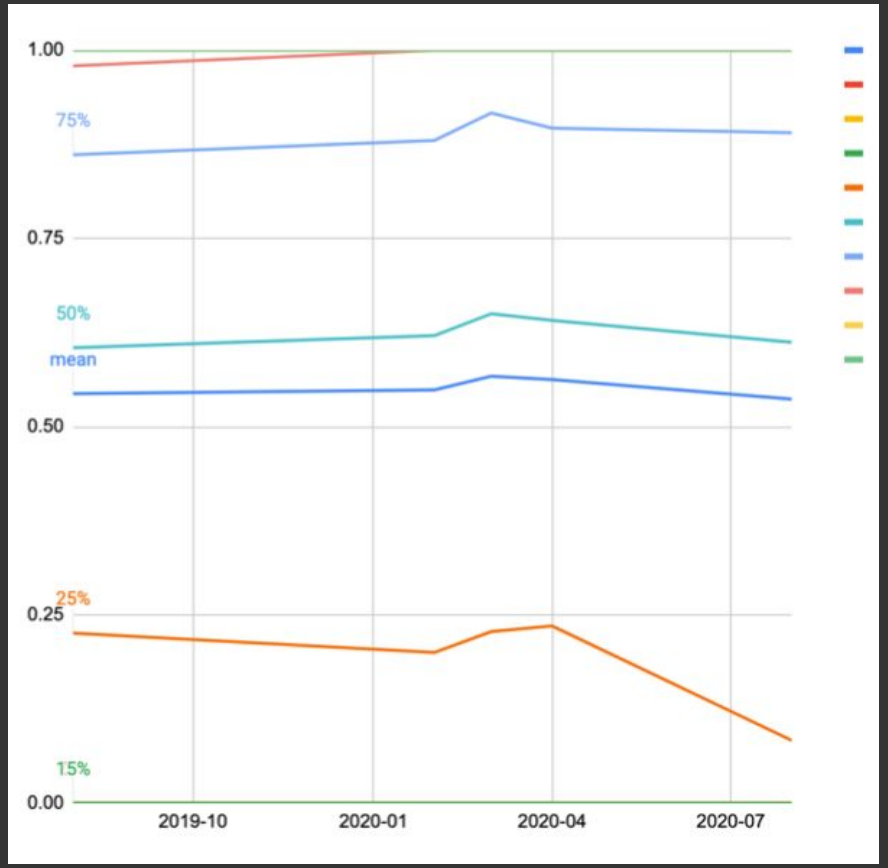


@IAmJerdog

Hypothesis: more tests written in March, driving up Duration. In April, a concerted effort on optimization

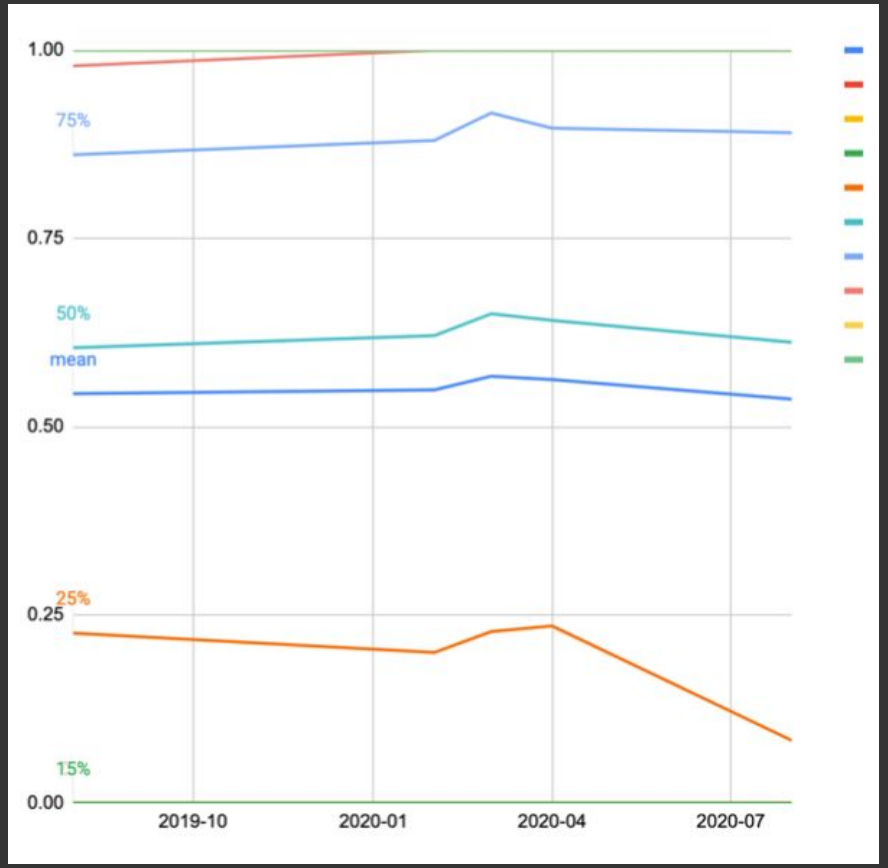
Success rate

Success rate in a global pandemic



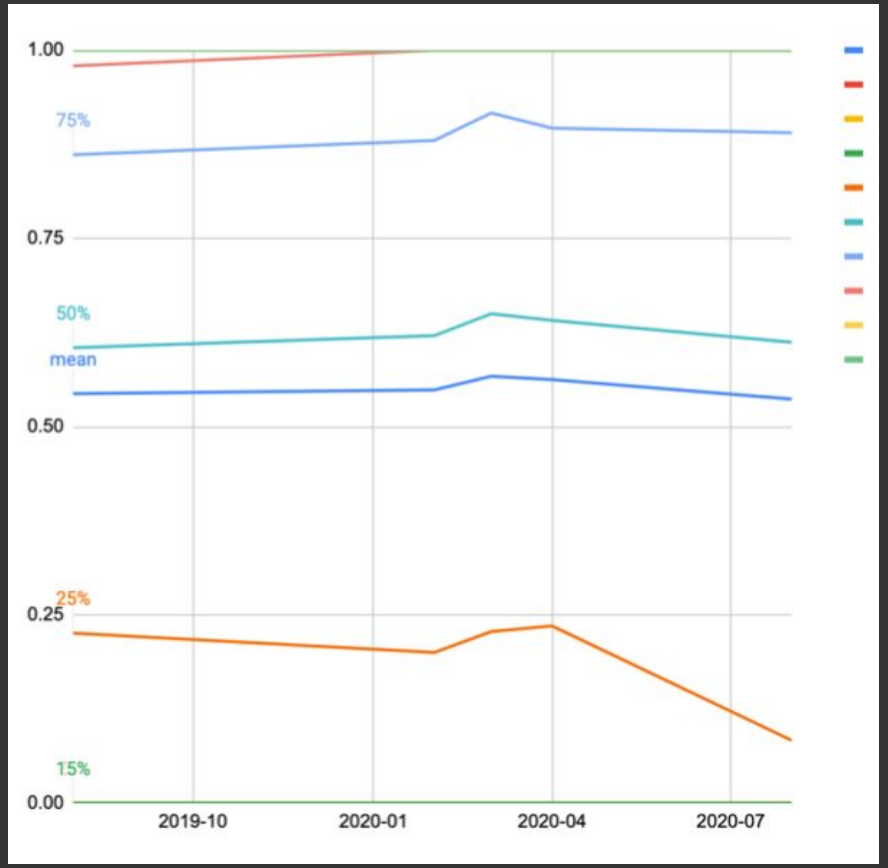
@IAmJerdog

Success rate in a global pandemic



@IAmJerdog

Success rate in a global pandemic

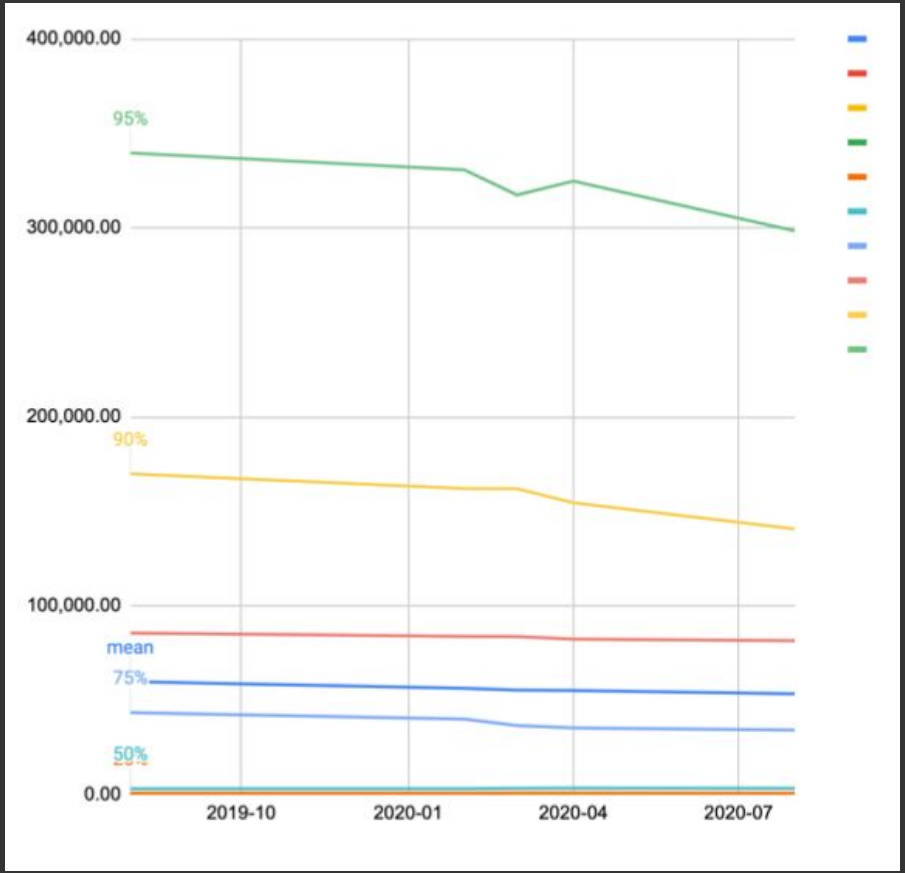


@IAmJerdog

Hypothesis: people working hard on core business stability

Recovery Time

Recovery time in a global pandemic



@IAmJerdog

Hypothesis: few distractions* working at home

Things that make you go 🤔

@IAmJerdog

Branch information

No significant change in default branch from **master**… yet.

Success Rate on default branch higher than on non-default

Duration on default branches *faster* at every percentile

Recovery Time lower on default branches at every percentile

What development practices definitively work?

Success Rate does not correlate with company size

@IAmJerdog

Duration is longest
for teams of one

@IAmJerdog

Recovery Time decreases with increased team size (up to 200)

Performance is better
with >1 contributor

Software is collaborative

Language by Throughput

1. Ruby
2. TypeScript
3. Go
4. Python
5. Kotlin
6. Elixir
7. Swift
8. HCL
9. JavaScript
10. TSQL
11. PHP
12. Java
13. C#
14. Jupyter Notebook
15. Shell
16. Vue
17. C++
18. HTML
19. CSS
20. Dockerfile

@IAmJerdog

Language by Success Rate

1. Vue
2. CSS
3. Shell
4. Dockerfile
5. TSQL
6. HTML
7. HCL
8. Go
9. TypeScript
10. JavaScript
11. Elixir
12. PHP
13. Jupyter Notebook
14. Python
15. Ruby
16. Java
17. Kotlin
18. C#
19. C++
20. Swift

@IAmJerdog

Language by fastest TTR

1. Go
2. JavaScript
3. Elicir
4. HCL
5. Shell
6. Python
7. TypeScript
8. CSS
9. C#
10. HTML
11. Vue
12. Jupyter Notebook
13. Kotlin
14. Java
15. Scala
16. Ruby
17. PHP
18. TSQL
19. Swift
20. C++

@IAmJerdog

Language by shortest duration

1. Shell
2. HCL
3. CSS
4. HTML
5. Gherkin
6. JavaScript
7. Vue
8. Go
9. Jupyter Notebook
10. Python
11. PHP
12. TypeScript
13. Java
14. Elixir
15. TSQL
16. Kotlin
17. Scala
18. Ruby
19. C++
20. Swift

@IAmJerdog

“Don’t deploy on Friday” is not a thing.

@IAmJerdog

“Don’t Deploy on Friday” is not a thing

- 70% less **Throughput** on weekends
- 11% less **Throughput** on Friday (UTC)
- 9% less **Throughput** on Monday (UTC)

2021/22 Sneak Peek

1. Workflows with 0 tests increase YoY, but decrease as total of all workflows
2. More deployments YoY
3. Change validation

2021/22 Sneak Peek

Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
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 committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

50th percentile on CircleCI fit into the “Elite performer” category on the 2021 State of DevOps report

@IAmJerdog

2021/22 Sneak Peek

Kubernetes usage with CI/CD has increased YoY

Full Report

https://circle.ci/ssd2020

Thank you.

For feedback and swag: **circle.ci/jeremy**

Timeline.jerdog.me

IAmJerdog

DEV jerdog

/in/jeremymeiss