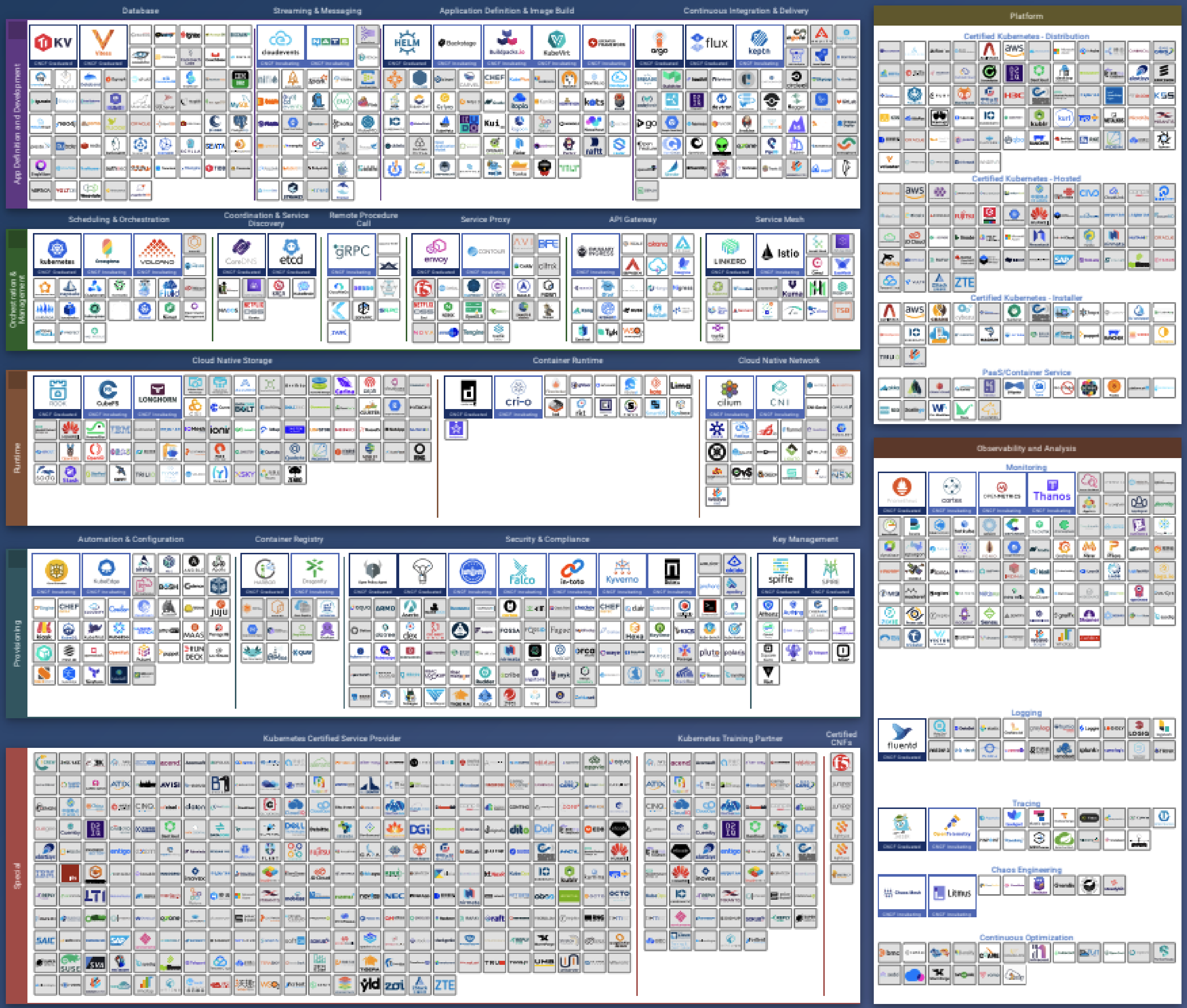
Metrics, High-performing teams, and the Holy Grail

{ }
HTML
</>

CNCF Cloud Native Landscape
1.0

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io



Jeremy Meiss

circleci

Director, DevRel & Community

@IAmJerdog

So back to the tech industry....

YOU SEEK THE HOLY GRAIL.

puppet labs
201
STATE
DevOps
REPORT
IT R
2016
State of DevOps Report
Presented by:
puppet
2017
State of DevOps Report
Presented by:
puppet + DORA
DEVOPS RESEARCH & ASSESSMENT
Sponsored by:
Hewlett Packard Enterprise
splunk>
amazon web services
Atlassian
WAVEFRONT by vmware
Deloitte.
Electric Cloud
2018
State of DevOps Report
Presented by:
puppet + splunk>
Sponsored by:
aws
cloudability
CYBERARK
Cognizant
diaxion
2019
State of DevOps Report
Presented by
puppet
circleci
splunk>
Sponsored by
ANITIAN
servicenow

Image: Risk Culture

CI/CD Benchmarks for high-performing teams

Duration

Mean time to recovery

Success rate

Throughput

So what does the data say?

Duration

the foundation of software engineering velocity, measures the average time in minutes required to move a unit of work through your pipeline

GIPHY

And There Was Much Rejoicing

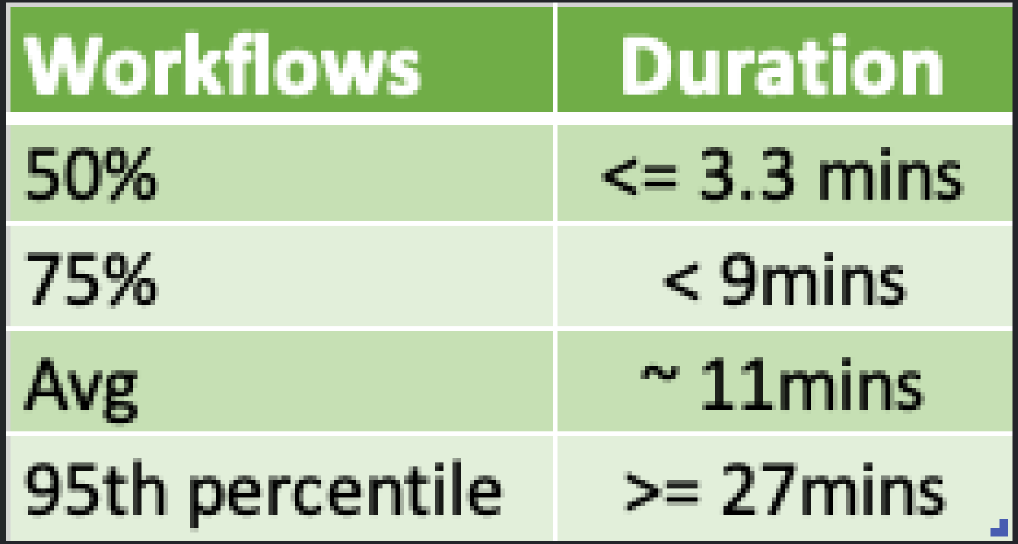
Duration Benchmark

<=10 minute builds

"a good rule of thumb is to keep your builds to no more than ten minutes. Many developers who use CI follow the practice of not moving on to the next task until their most recent checkin integrates successfully. Therefore, builds taking longer than ten minutes can interrupt their flow."

-- Paul M. Duvall (2007). *Continuous Integration: Improving Software Quality and Reducing Risk*

Duration: What the data shows



Workflows	Duration
50%	<= 3.3 mins
75%	< 9mins
Avg	~ 11mins
95th percentile	>= 27mins

Benchmark: 5-10mins

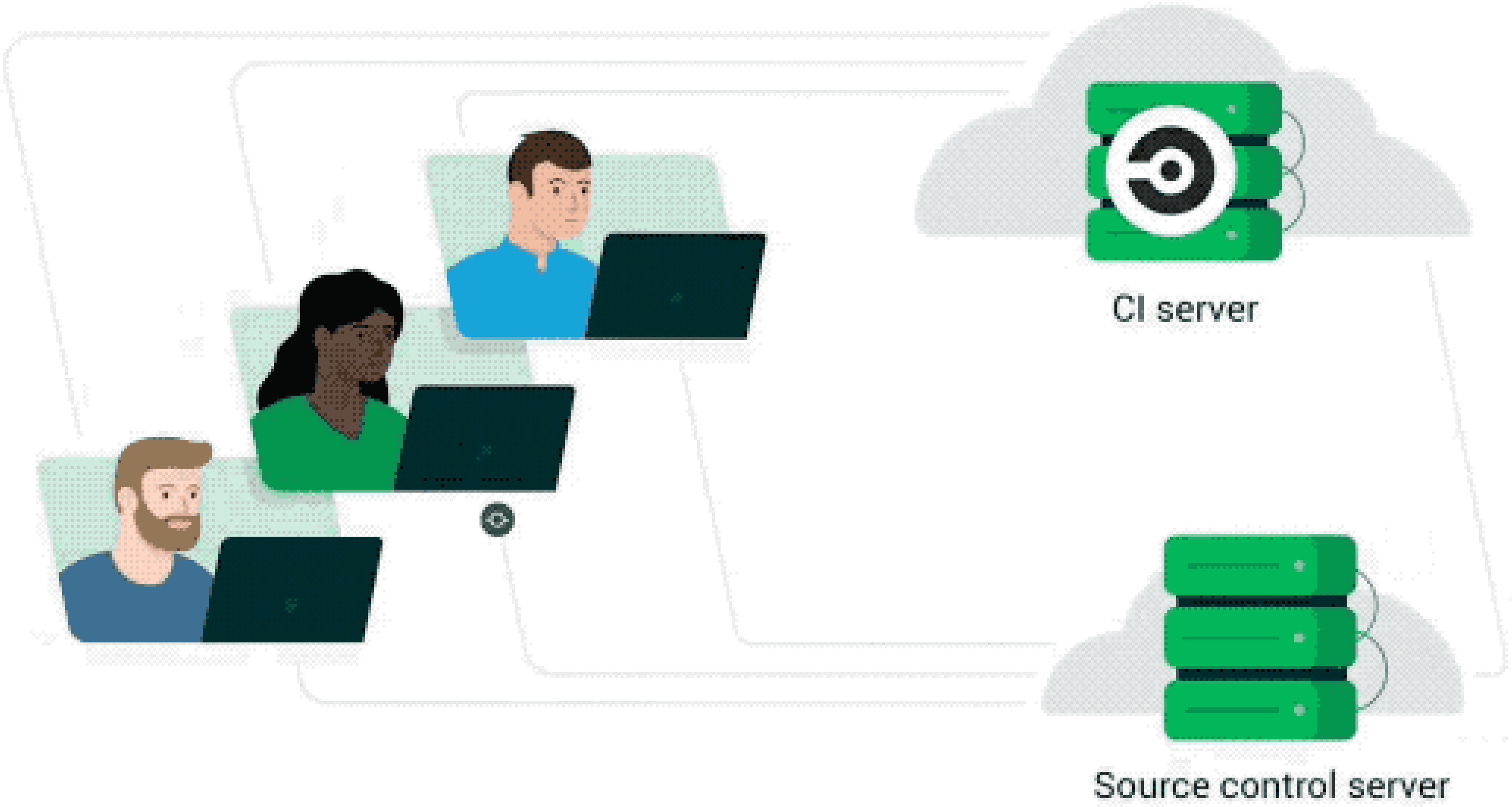
Improving test coverage

- Add unit, integration, UI, and end-to-end testing across all app layers
- Incorporate code coverage tools into pipelines to identify inadequate testing
- Include static and dynamic security scans to catch vulnerabilities
- Incorporate TDD practices by writing tests during design phase

Mean time to Recovery

the average time required to go from a failed build signal to a successful pipeline run

Mean time to recovery is indicative of resilience


CI server
Source control server

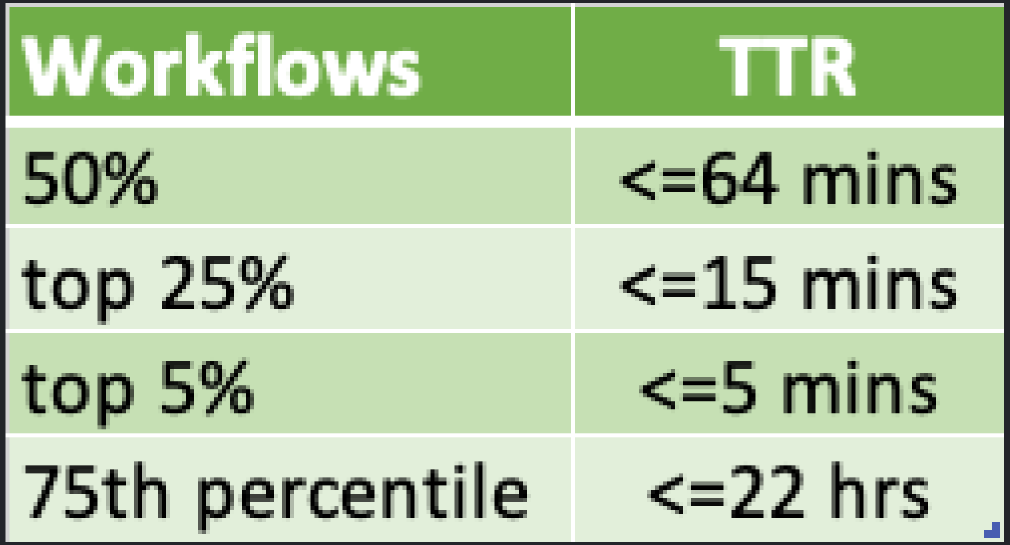
"A key part of doing a continuous build is that if the mainline build fails, it needs to be fixed right away. The whole point of working with CI is that you're always developing on a known stable base."

-- Fowler, Martin. "Continuous Integration." Web blog post. *MartinFowler.com*. 1 May 2006. Web.

MTTR Benchmark

<=60min MTTR on
default branches

MTTR: What the data shows



Workflows	TTR
50%	<=64 mins
top 25%	<=15 mins
top 5%	<=5 mins
75th percentile	<=22 hrs

Benchmark: 60mins

Treat your default branch as the lifeblood of your project

Getting to faster recovery times

- Treat your **default branch as the lifeblood** of your project
- Set up **instant alerts for failed builds** using services like Slack, Twilio, or Pagerduty.
- Write **clear, informative error messages** for your tests that allow you to quickly diagnose the problem and focus your efforts in the right place.
- **SSH into the failed build machine to debug** in the remote test environment. Doing so gives you access to valuable troubleshooting resources, including log files, running processes, and directory paths.

Success Rate

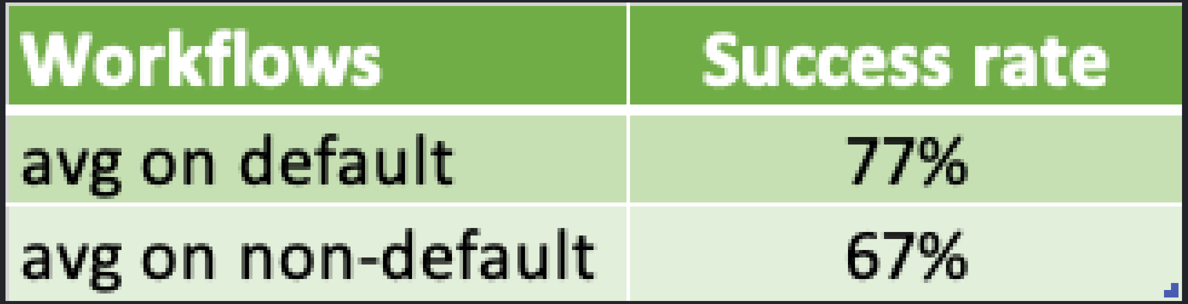
number of passing runs divided by the total number of runs over a period of time

now go away...
...or i will taunt you a second time!

Success Rate Benchmark

90%+ Success rate on default branches

Success rate: What the data shows

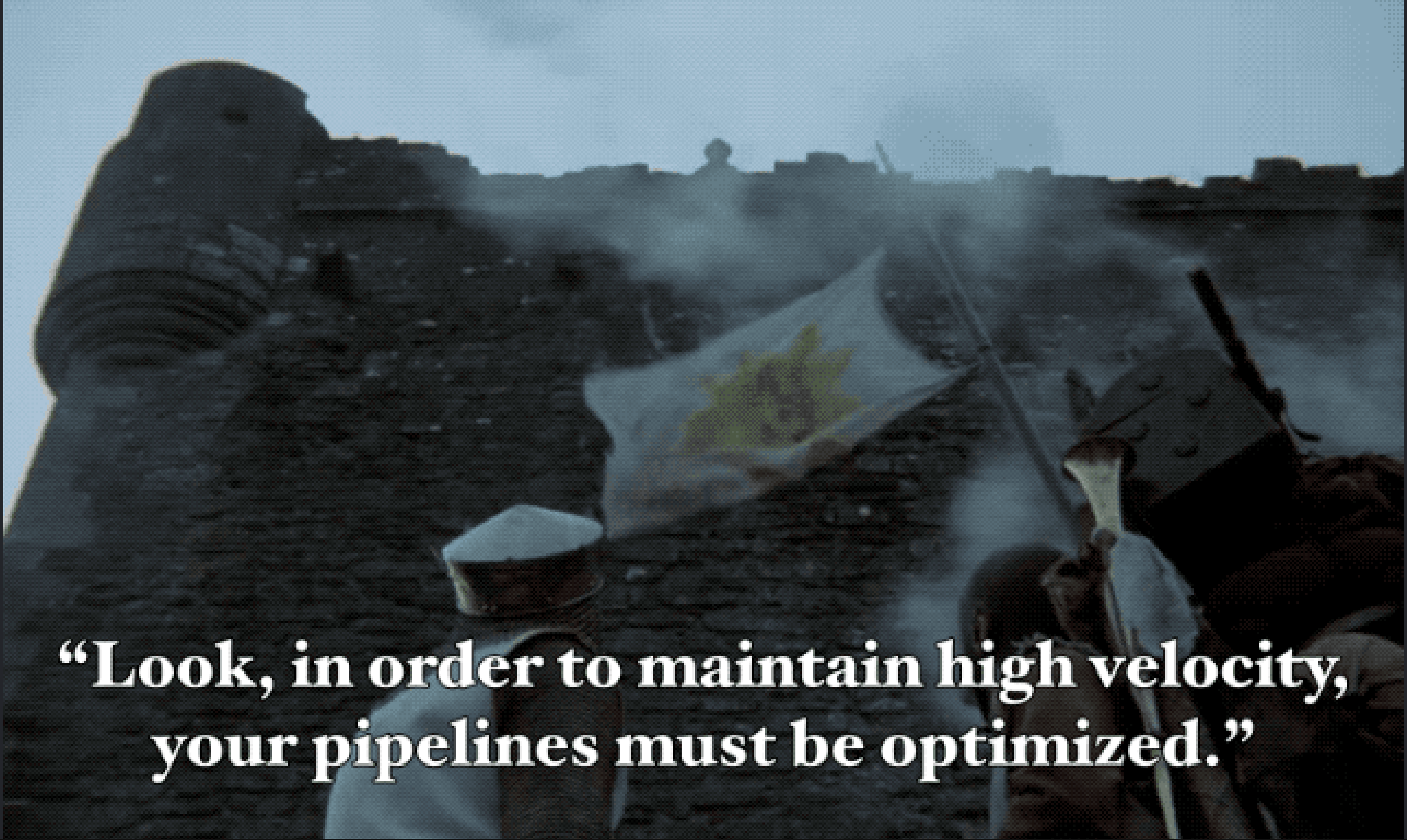


Workflows	Success rate
avg on default	77%
avg on non-default	67%

Benchmark: 90%+ on default

Throughput

average number of workflow runs that an organization completes on a given project per day



“Look, in order to maintain high velocity, your pipelines must be optimized.”

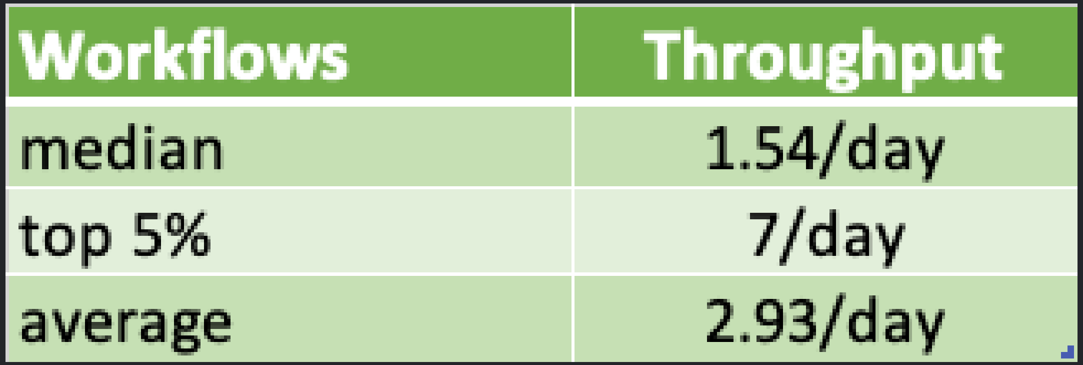
It's only a model.

So what Throughput is ideal?

Throughput Benchmark

It depends.

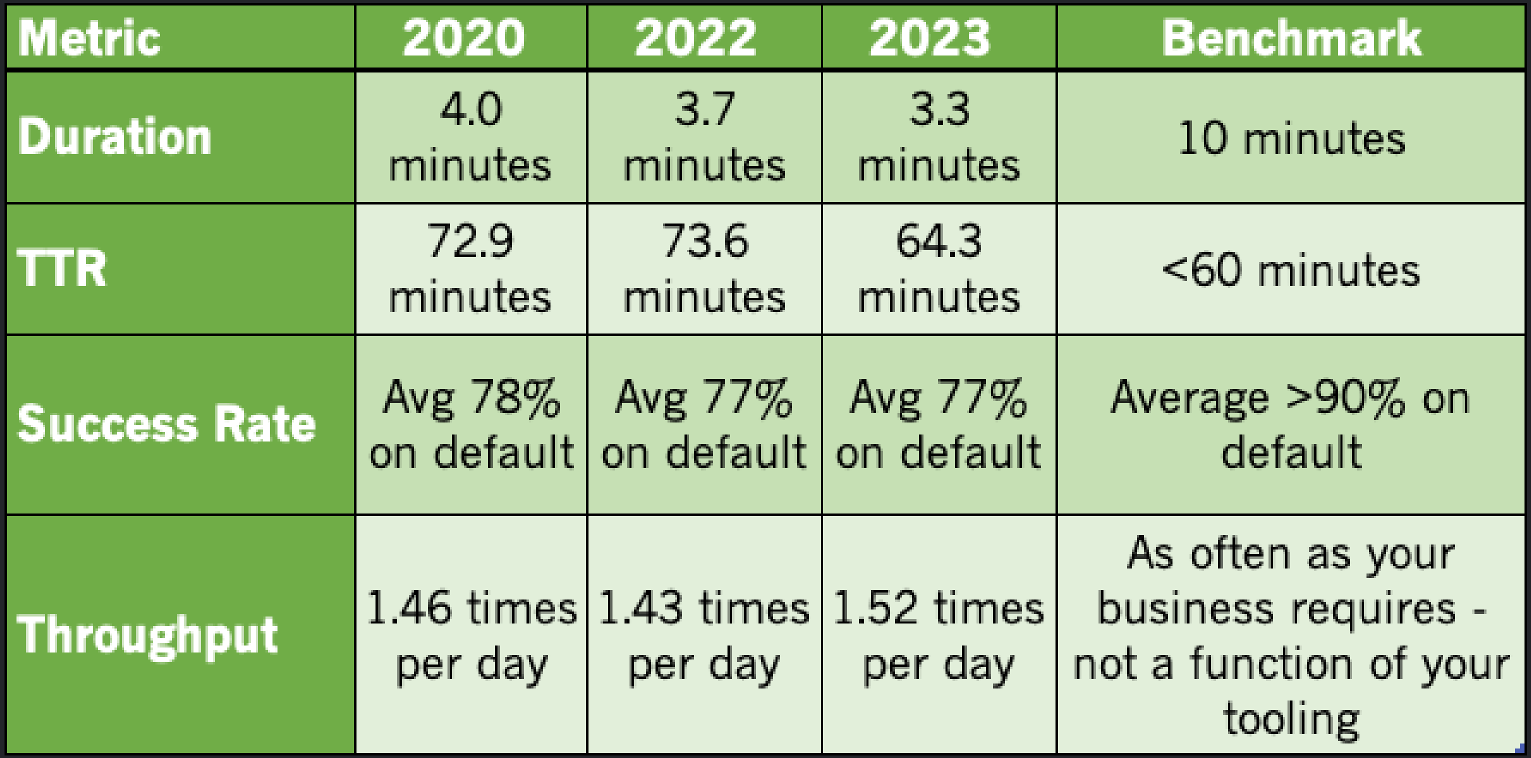
Throughput: What the data shows



Workflows	Throughput
median	1.54/day
top 5%	7/day
average	2.93/day

Benchmark: at the speed of your business

High-Performing Teams in 2023



Metric	2020	2022	2023	Benchmark
Duration	4.0 minutes	3.7 minutes	3.3 minutes	10 minutes
TTR	72.9 minutes	73.6 minutes	64.3 minutes	<60 minutes
Success Rate	Avg 78% on default	Avg 77% on default	Avg 77% on default	Average >90% on default
Throughput	1.46 times per day	1.43 times per day	1.52 times per day	As often as your business requires - not a function of your tooling

Platform teams and their impact

The impact of Platform Teams

Duration

- Identify and eliminate impediments to developer velocity
- Set guardrails and enforce quality standards across projects
- Standardize test suites and CI pipeline configs, i.e. shareable config templates and policies

MTTR

- Set up effective monitoring and alerting systems, and track recovery time
- Config- and Infrastructure-as-Code tools limit potential for misconfig errors
- Actively monitor, streamline, and parallelize pipelines across the org

Success rate

- Look at MTTR and shorten recovery time first
- Set a baseline success rate, then aim for continuous improvement
- Be mindful of patterns and influence of external factors

Throughput

- Map goals to reality of internal and external business situations
- Capture a baseline, monitor for deviations
- Alleviate as much developer cognitive load from day-to-day work

2020 Report

https://circle.ci/ssd2020

Full 2022 Report

https://circle.ci/ssd2022

Thank You.

For feedback and swag: circle.ci/jeremy

timeline.jerdog.me

IAmJerdog

DEV jerdog

in /in/jeremymeiss

@jerdog@hachyderm.io