



How do you know your DevEx Efforts are working?

Michael Stahnke
VP Engineering at Flox (flox.dev)
@stahnma

Career



9 Years

CATERPILLAR[®]

8 Years

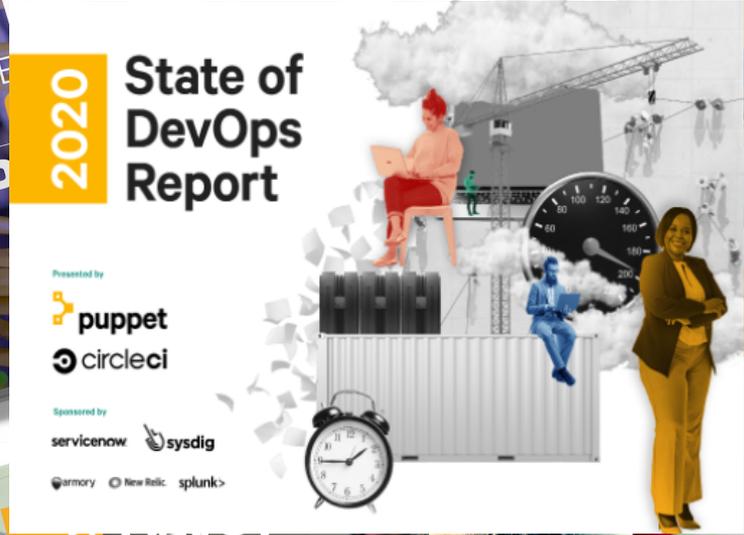
 **puppet**

4 Years

 **circleci**

About a Year

 **FLOX**



The 2020 State of Software Delivery

2020

Data-Backed Benchmarks for Engineering Teams

By Ron Powell and Michael Stahnke

BENCHMARK #1

5-10 min

Range for workflow durations, on average



BENCHMARK #2

<1 hour

Timeline for fixing or reverting any failed run



BENCHMARK #3

>90%

Success rates for the default branch



BENCHMARK #4

1+ times/day

Rate of deployment; higher as the business requires



circleci

The 2022 State of Software Delivery

2022

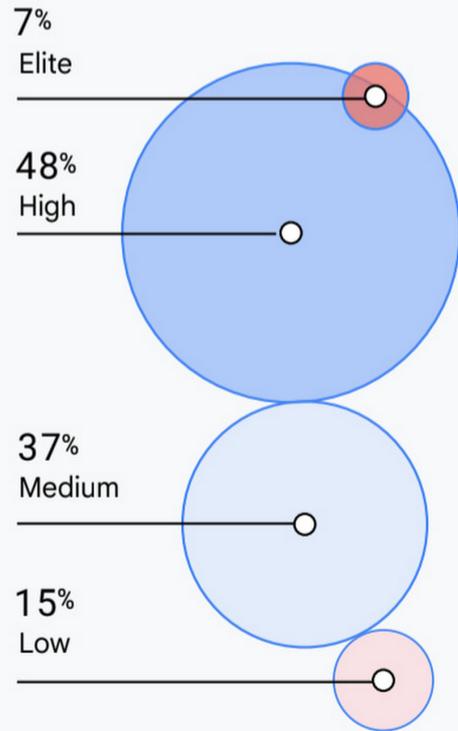
by Ron Powell



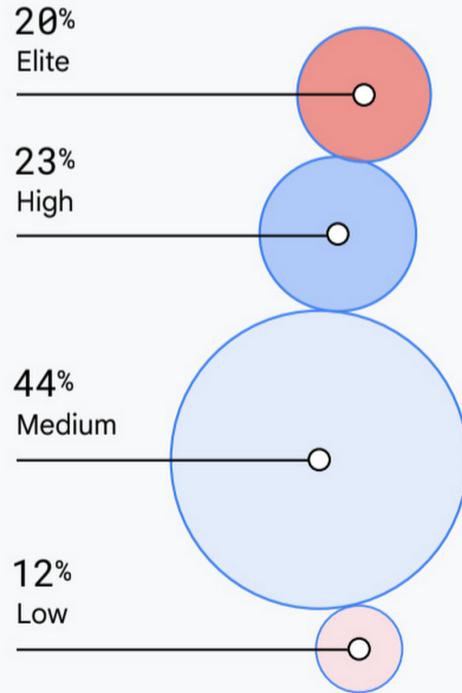
Uplevel

Software delivery performance metric	Elite	High	Medium	Low
<p>🔄 Deployment frequency</p> <p>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?</p>	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
<p>⌛ Lead time for changes</p> <p>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)?</p>	Less than one hour	Between one day and one week	Between one month and six months	More than six months
<p>🕒 Time to restore service</p> <p>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)?</p>	Less than one hour	Less than one day	Between one day and one week	More than six months
<p>⚠️ Change failure rate</p> <p>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)?</p>	0%-15%	16%-30%	16%-30%	16%-30%

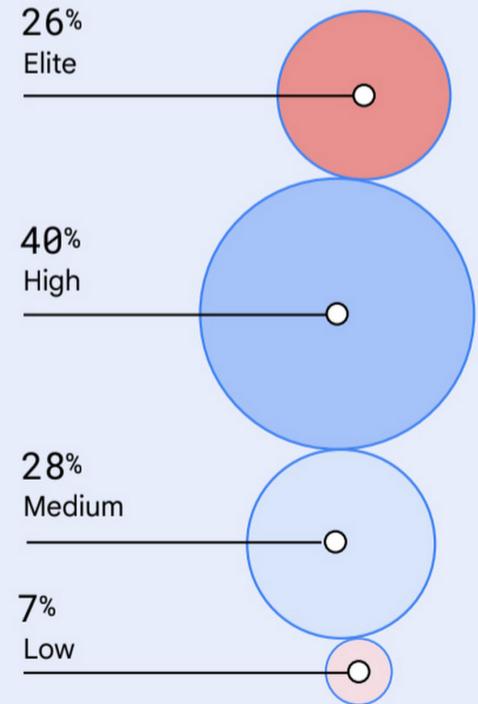
2018



2019

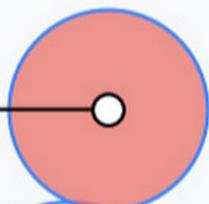


2021

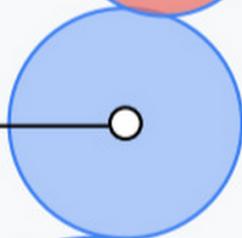


2019

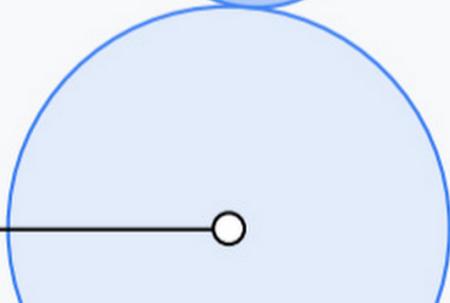
20%
Elite



23%
High

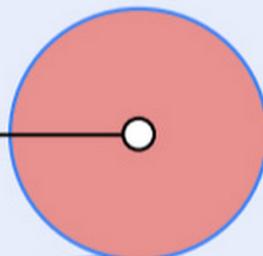


44%
Medium

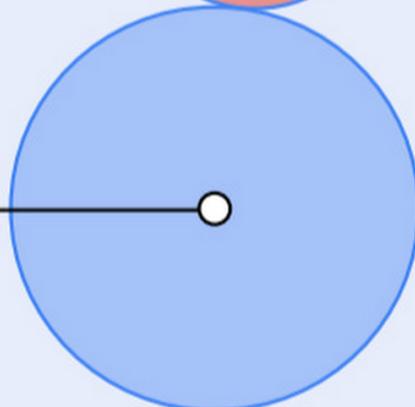


2021

26%
Elite



40%
High



28%
Medium



2021

We are here.



26%
Elite



40%
High



Availability

Cost for Downtime

COGS

Cost for Services in support of Gross Margin

Security

Attach Rate and dollar value for compliance docs and security questionnaires

Developer Productivity

Hiring Tradeoffs, time to value, throughput

Developer Sentiment

Developer Adoption



Developer Productivity

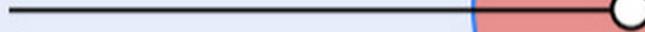
Hiring Tradeoffs, time to
value, throughput

2021

We are here.



26%
Elite



40%
High



Do I believe this?

Software delivery performance metric	Elite	High	Medium	Low
<p>🔄 Deployment frequency</p> <p>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?</p>	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
<p>⌛ Lead time for changes</p> <p>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)?</p>	Less than one hour	Between one day and one week	Between one month and six months	More than six months
<p>🕒 Time to restore service</p> <p>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)?</p>	Less than one hour	Less than one day	Between one day and one week	More than six months
<p>⚠️ Change failure rate</p> <p>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)?</p>	0%-15%	16%-30%	16%-30%	16%-30%

Software delivery performance metric	Elite	High	Medium	Low
<p>🔄 Deployment frequency</p> <p>For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?</p>	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
<p>⌚ Lead time for changes</p> <p>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)?</p>	Less than one hour	Between one day and one week	Between one month and six months	More than six months
<p>🕒 Time to restore service</p> <p>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)?</p>	Less than one hour	Less than one day	Between one day and one week	More than six months
<p>⚠️ Change failure rate</p> <p>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)?</p>	0%-15%	16%-30%	16%-30%	16%-30%

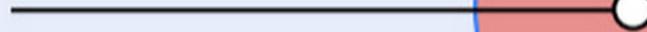
Where have I found my orgs historically?

2021

We are here.



26%
Elite



40%
High



I don't feel elite.

Why don't I feel elite?

What truly is elite?

Moving fast with confidence.

Move from IC reported to aggregate

Move from reported to derived



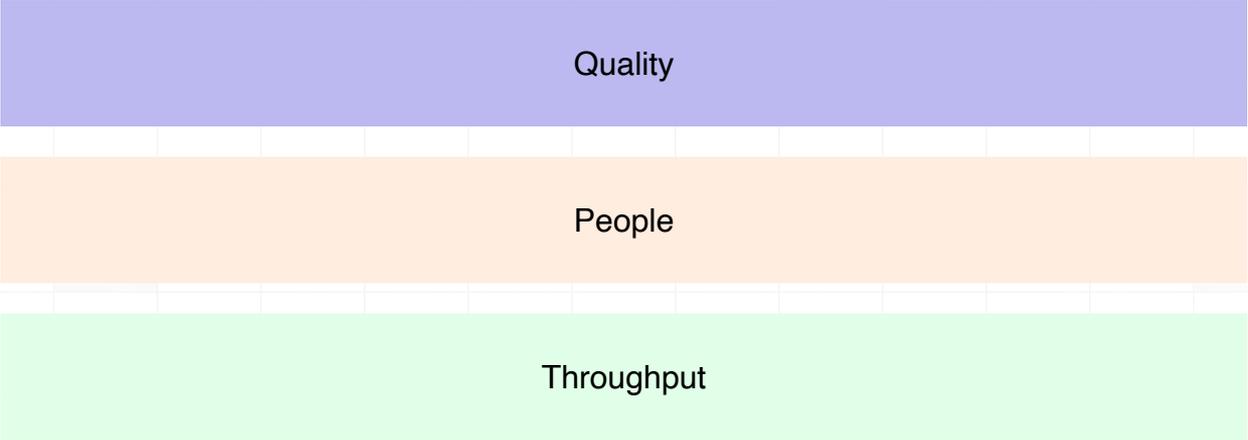
What are the categories of things I care about?

Is engineering throughput good?

Is that throughput meaningful?

**If I add more engineers, will we go faster?
Can I prove that?**

Metric Thoughts Layer Cake



Quality

People

Throughput

deployments / week

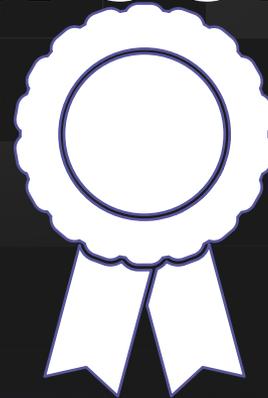
Throughput Measurements

deployments / week

Does adding more developers make things go faster?

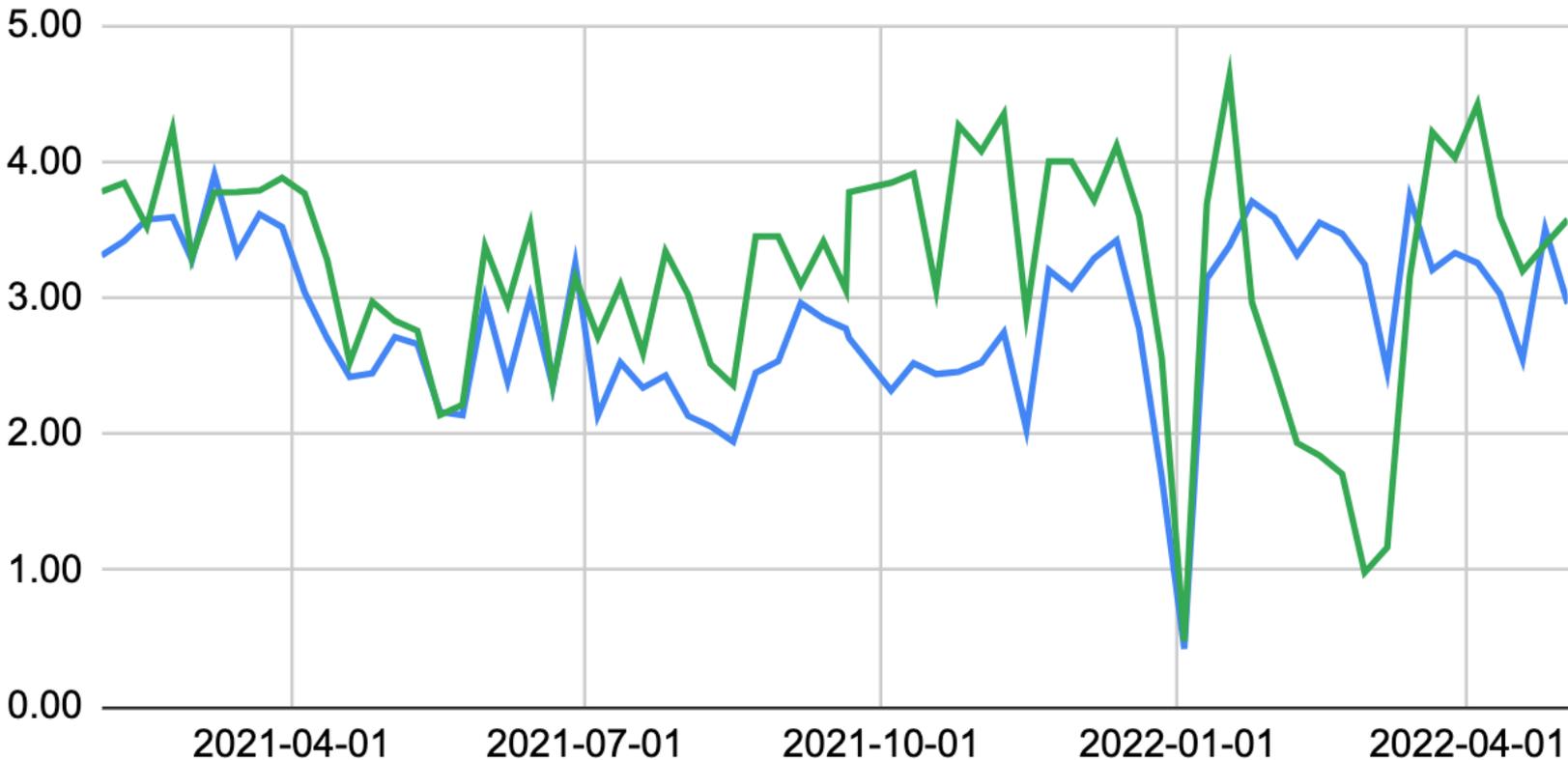
deploys / dev / week

deploys / dev / week



Weekly PRs & Deploys per Engineer

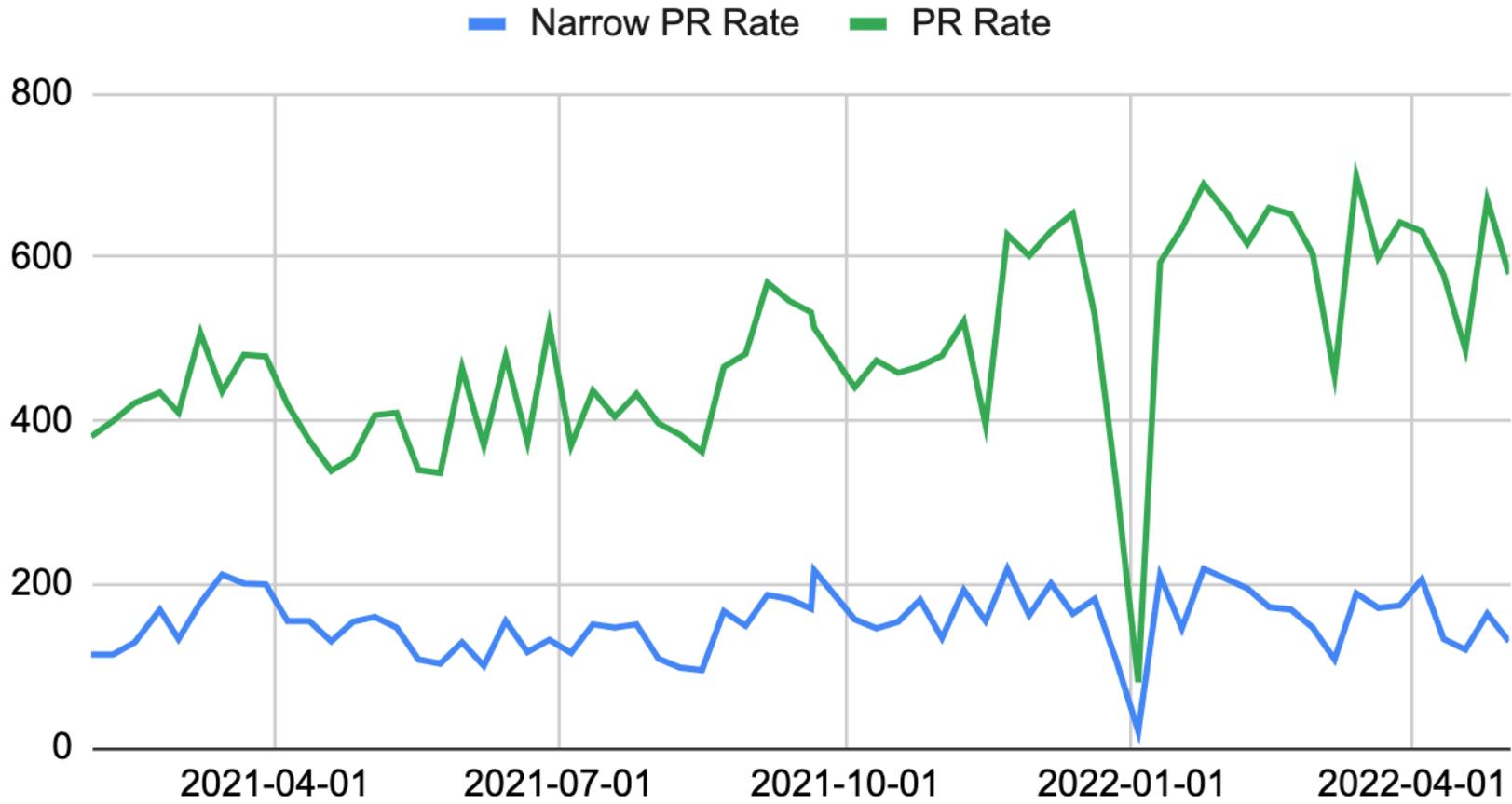
PR/Engineer Deployments/Engineer



ratio of distraction action

narrow vs wide work

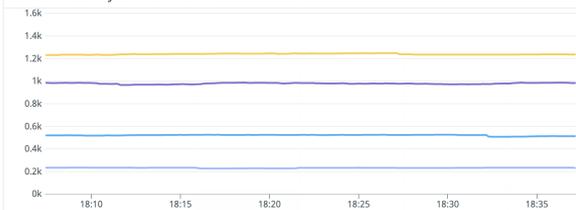
Weekly PR & Narrow PR Rate



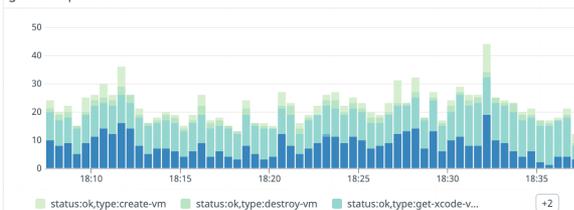
People

Are our systems easy to work on?

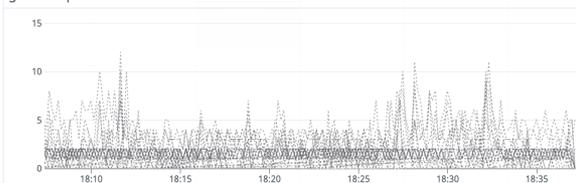
Available VMs by Resource Class



gRPC Responses



gRPC Request distribution



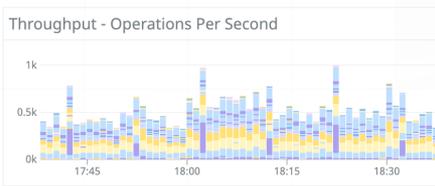
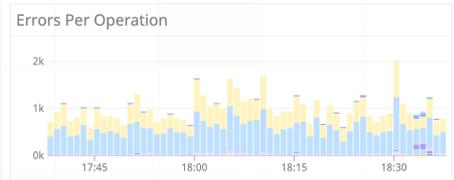
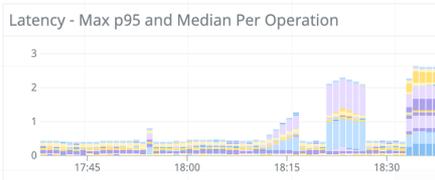
VM Boot Success Ratio by Resource Class



gRPC API

Success Rate (All Endpoints)

TYPE	SUCCESS	↓ AVG...	BAD_RE...	VOLUME	UPTIME...
get-build	3.39k	98.52	N/A	3.39k	100
get-wor...	2.72k	66.79	N/A	2.72k	99.96
get-proj...	4.02k	66.44	N/A	4.41k	91.15
get-wor...	2.41k	62.77	N/A	2.41k	100
get-proj...	1.03k	34.19	N/A	1.42k	72.71
usage...	9.29k	26.02	N/A	9.29k	100
get-buil...	910	24.19	N/A	910	100
get-feat...	1k	21.07	N/A	1k	100
get-con...	1.13k	19.45	N/A	1.18k	95.67



fungible engineering?

ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS



ESSAYS ON SOFTWARE ENGINEERING

THE MYTHICAL MAN-MONTH

FREDERICK P. BROOKS, JR.

normalization

**% of code provided
by the platform.**

KPI	2022-04-11	2022-04-18	2022-04-25	2022-05-02
% of services using clj-parent	100.00%	100.00%	100.00%	100.00%
% of services using a recent Backplane release	26.92%	26.92	78.85%	78.85%
% of services using clj-parent which are on a recent Backplane release	26.92%	26.92	78.85%	78.85%
% of services with tracing enabled	100.00%	100.00%	100.00%	100.00%
% of services with additional tracing	82.69%	82.69%	82.69%	82.69%

How quickly are they productive?

5th contribution (PR)

deploys / dev / week



Quality

successful deployments / week

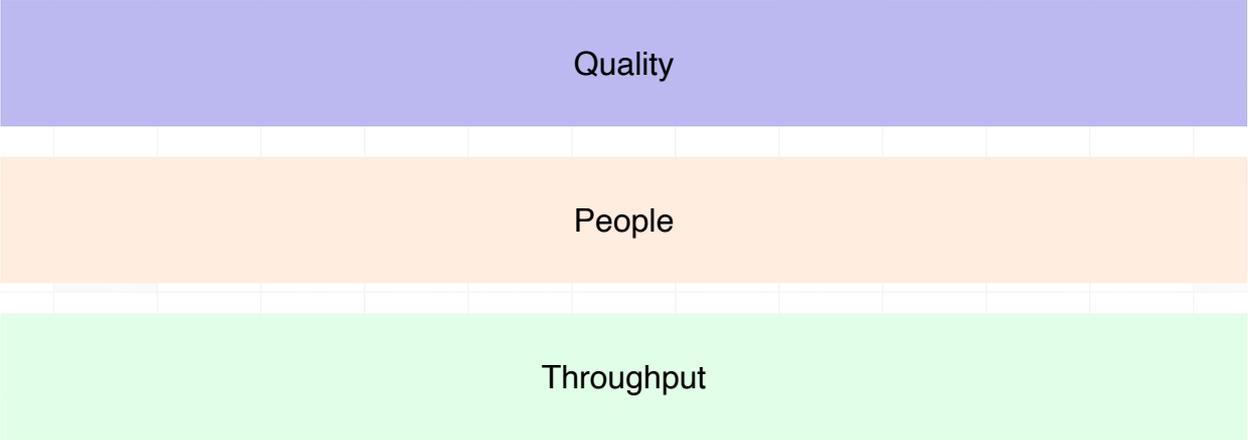
rollbacks

(or roll forwards)

**understanding changes happening all
around you**

	03-Apr-2022	10-Apr-2022	17-Apr-2022	24-Apr-2022	May	2-May-2022
Release Frequency (deployment per dev per week)	4.61	3.67	3.25	3.47		3.75
Releases	858	687	611	648		701
Rollback Rate	9	14	5	7		8
Success Rate	98.95%	97.96%	99.18%	98.92%		98.86%

Metric Thoughts Layer Cake



Quality

People

Throughput

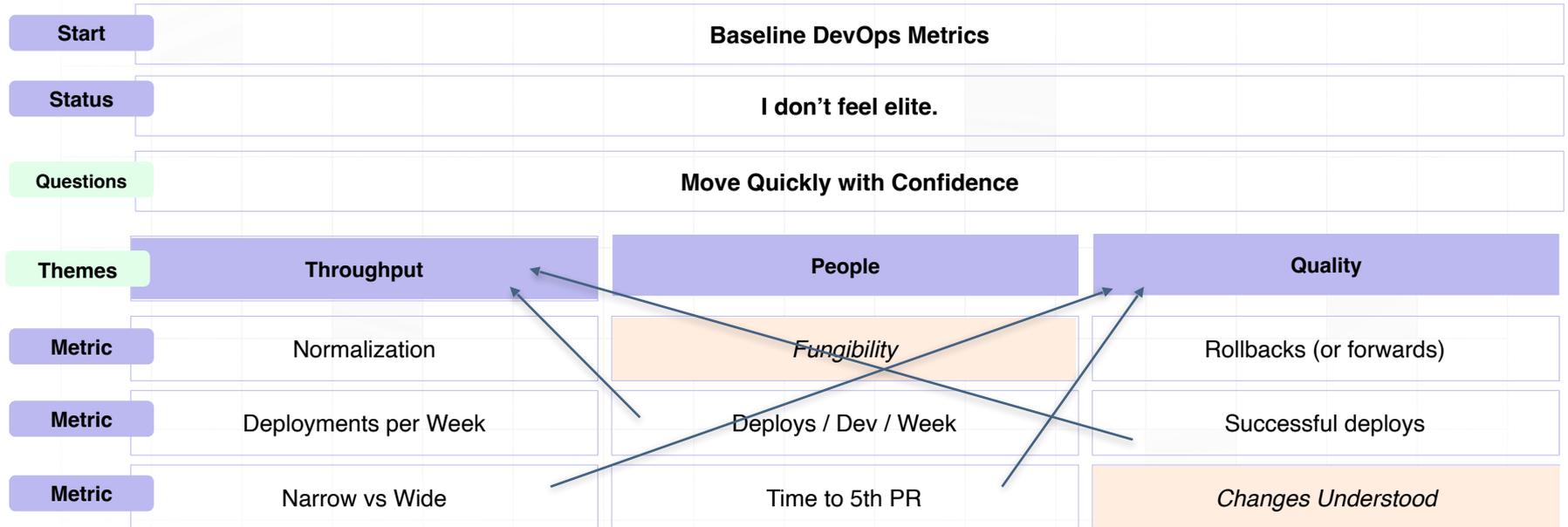
Mental Model for Metrics

Start	Baseline DevOps Metrics		
Status	I don't feel elite.		
Questions	Move Quickly with Confidence		
Themes	Throughput	People	Quality
Metric	Normalization	Fungibility	Rollbacks (or forwards)
Metric	Deployments per Week	Deploys / Dev / Week	Successful deploys
Metric	Narrow vs Wide	Time to 5th PR	Changes Understood

Mental Model for Metrics

Start	Baseline DevOps Metrics		
Status	I don't feel elite.		
Questions	Move Quickly with Confidence		
Themes	Throughput	People	Quality
Metric	Normalization	<i>Fungibility</i>	Rollbacks (or forwards)
Metric	Deployments per Week	Deploys / Dev / Week	Successful deploys
Metric	Narrow vs Wide	Time to 5th PR	<i>Changes Understood</i>

Mental Model for Metrics



Summary

- What do you want your metrics to tell you?
- Can you correlate the movement with intentional action and improved outcomes?
- Do you have baseline metrics already?
- How have you joined these up with product management?
- Deploys / Dev / Week is the `_one_` I would start with



Twitter's Engineering Effectiveness

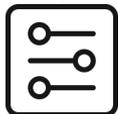
Let 1,000 Flowers Bloom - <http://www.gigamonkeys.com/flowers/>



The SPACE of developer Productivity.

Research paper from
Nicole Forsgren, GitHub
Margaret-Anne Storey, University of Victoria
Chandra Maddila, Thomas Zimmermann, Brian Houck,
and Jenna Butler, Microsoft Research

<https://queue.acm.org/detail.cfm?id=3454124>



GitLab MR Rate

<https://about.gitlab.com/handbook/engineering/merge-request-rate/>



flox.dev