Bell the “Chaotic Cat” with SRE

/mandeepsu

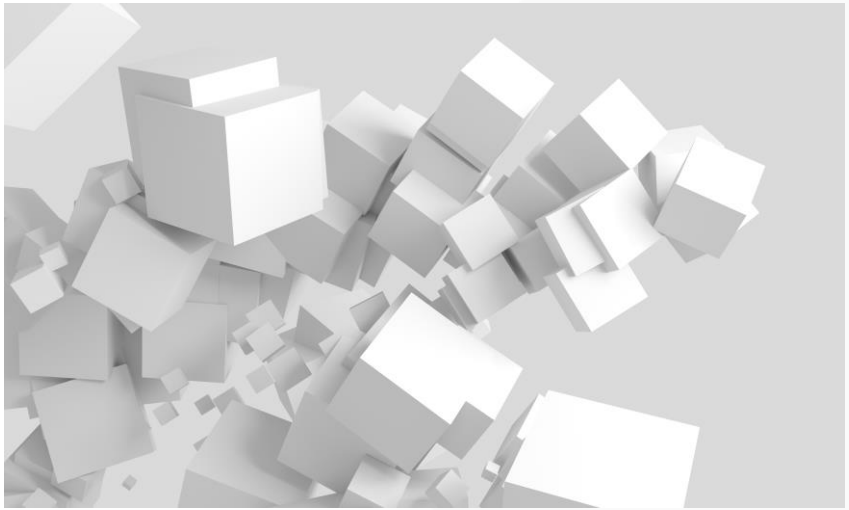
Vision

To understand how SRE core fundamentals can help address unintentional chaos in software / product development lifecycle

Outcome

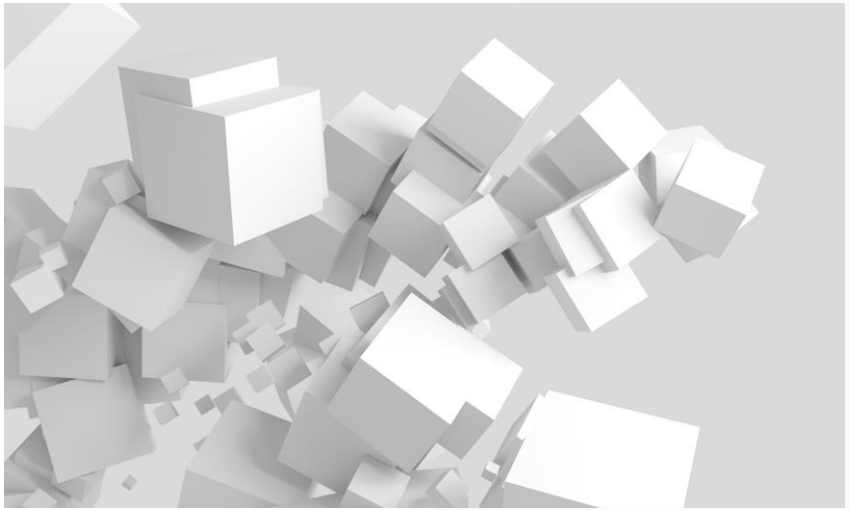
Understanding of how SRE core fundamentals can align with product development lifecycle

Chaos Engineering



Intentional chaos in production to identify opportunities to improve resiliency

Chaos Engineering



Intentional chaos in production to identify opportunities to improve resiliency

But what about **un**intentional chaos in product development lifecycle?

3 Themes – 1/3

Speed

When it comes to faster time to market and a higher number of deployments per day / week / month, what are we chasing? How many number of deployments are good enough? Do our customers even care whether we are in “Elite” category?

3 Themes – 2/3

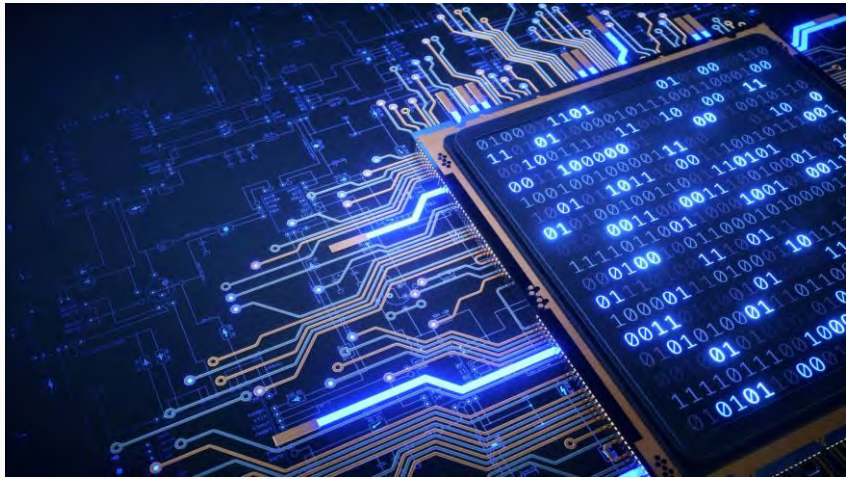
Vendor / Upstream / Downstream Dependencies

For feature rollouts and production incidents, how much time we spend collaborating with other teams to ensure that “right team” is engaged and “everyone is on the same page”?

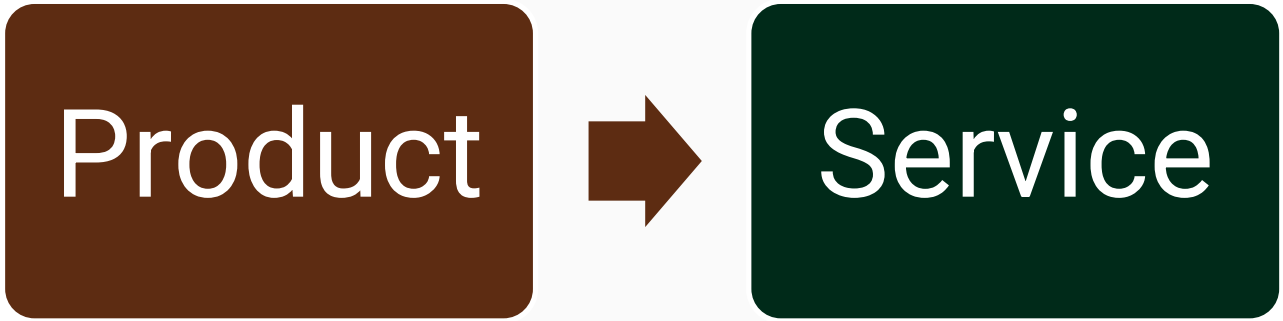
3 Themes – 3/3

Measuring Everything

Should we? If not, how do we know what not to measure?

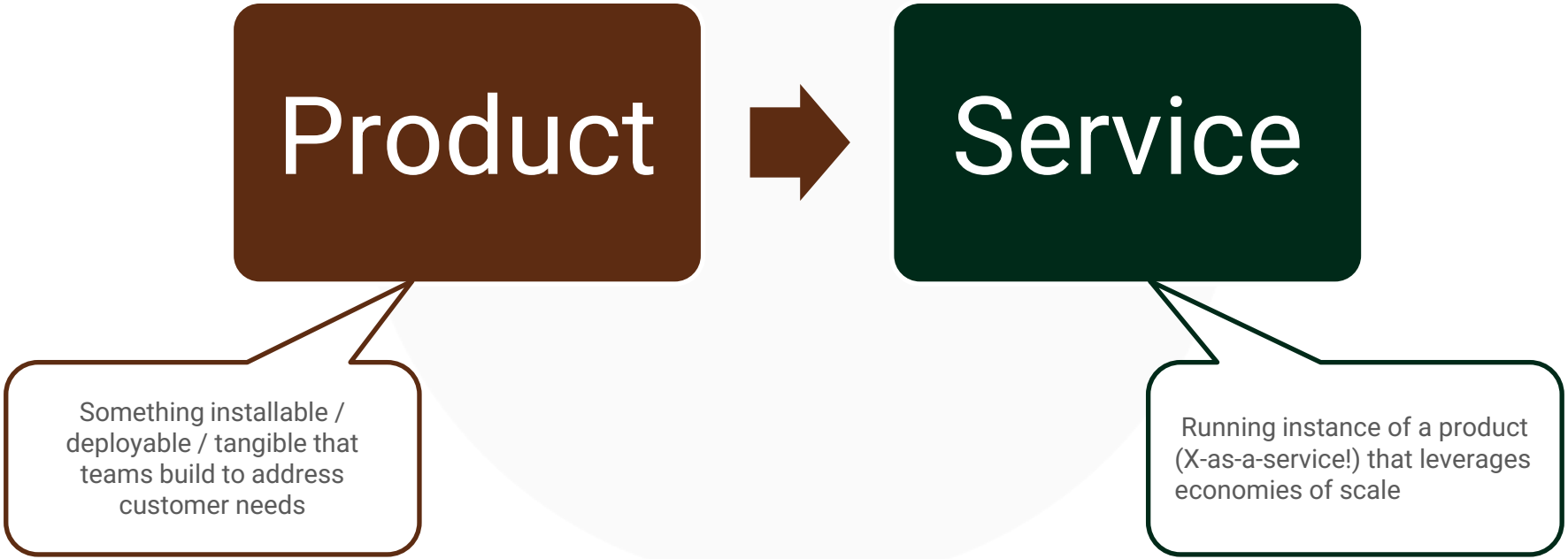


Product vs Service

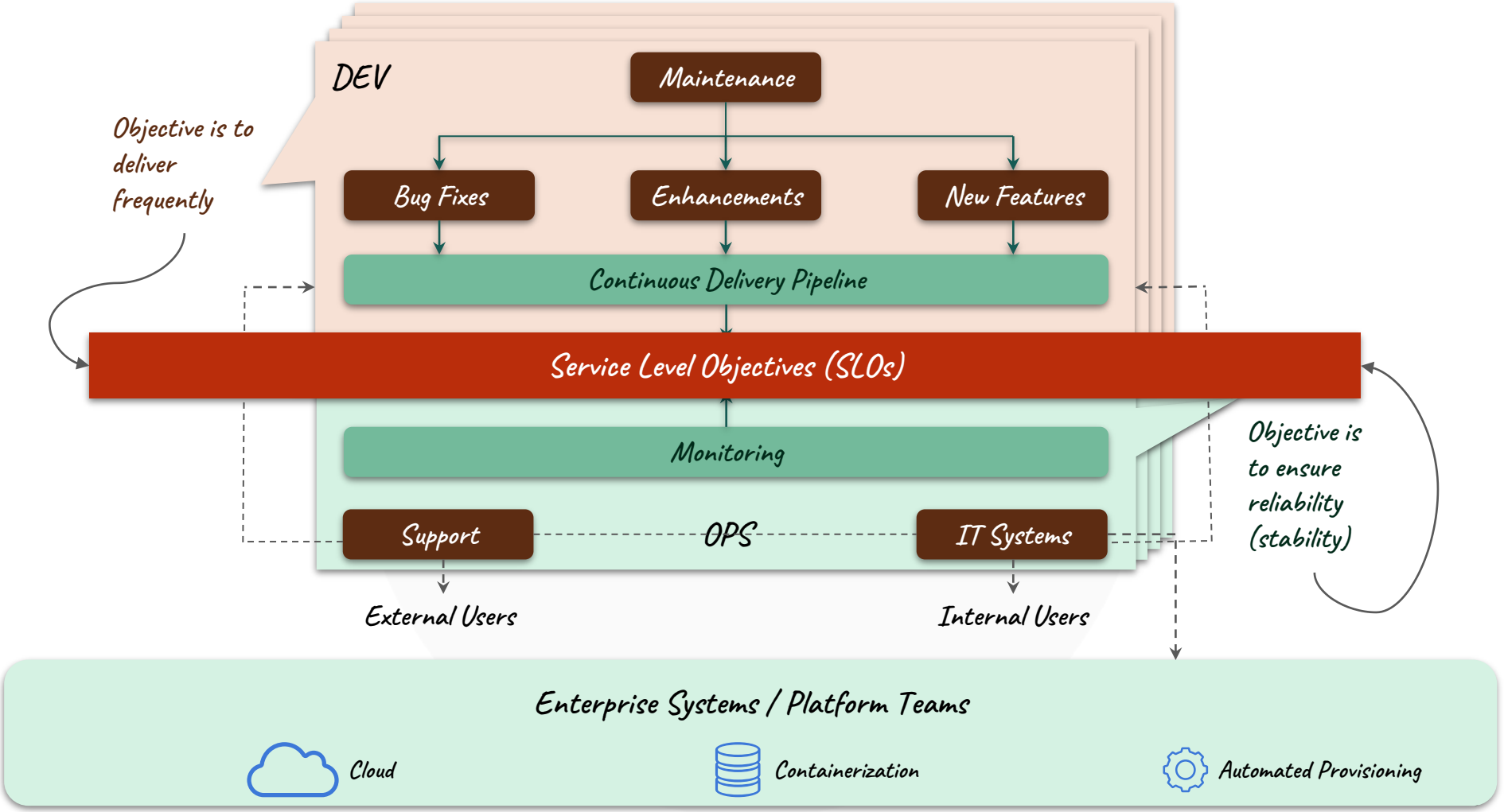


Product Reliability?

Product vs Service



How does SRE help?


DEV
Maintenance
Objective is to deliver frequently
Bug Fixes
Enhancements
New Features
Continuous Delivery Pipeline
Service Level Objectives (SLOs)
Monitoring
Support
OPS
IT Systems
External Users
Internal Users
Objective is to ensure reliability (stability)
Enterprise Systems / Platform Teams
Cloud
Containerization
Automated Provisioning

What is SLO?

In simplest terms, an SLO is a balancig lever.

Service Level Indicator (SLI)

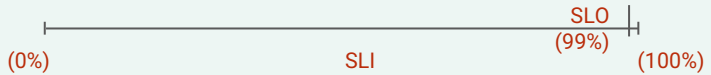
SLI - a carefully defined quantitative measure of some aspect of the level of service that is provided.

User Expectations	SLI (Availability)
Login request should complete successfully	**Ratio** of login requests that are successful

Service Level Objective (SLO)

SLO - a target value, across a time window, for a service level that is measured by an SLI.

SLI (Availability)	SLO (Availability)
Ratio of login requests that are successful	**99%** of login requests (**averaged over previous 30 days**) should be successful



HOW to Implement SLO?

SLI ⇐⇒ SLO

Work About Work (Toil?)

How can product teams free up their unnecessarily consumed bandwidth?

HOW is Toil Different from Overheads?

Toil

- Manual
- Repetitive
- Automatable
- Tactical
- No enduring value
- $O^{(n)}$ with service growth

Overhead

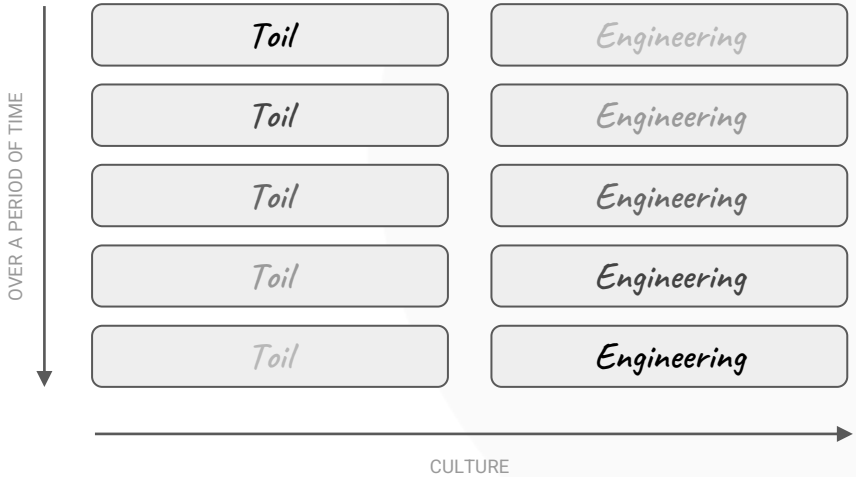
- Cognitive; requires conscious attention
- Could be repetitive
- Not automatable
- Strategic
- Long term enduring value
- Not necessarily related to service management

Examples of toil (“Montra”) that can be addressed to release developer bandwidth:

- Setting up development / testing environments for different scenarios (How do we reproduce this?)
- Upgrading libraries / APIs to address e.g. vulnerability issues (All apps to implement TLS 1.3 by Q2)
- General work about work (How fast can users log in to app X? What’s the latency trend? RAG? One pager?)

HOW to ~~Eliminate~~ Reduce Toil?

By engineering - not just automation!

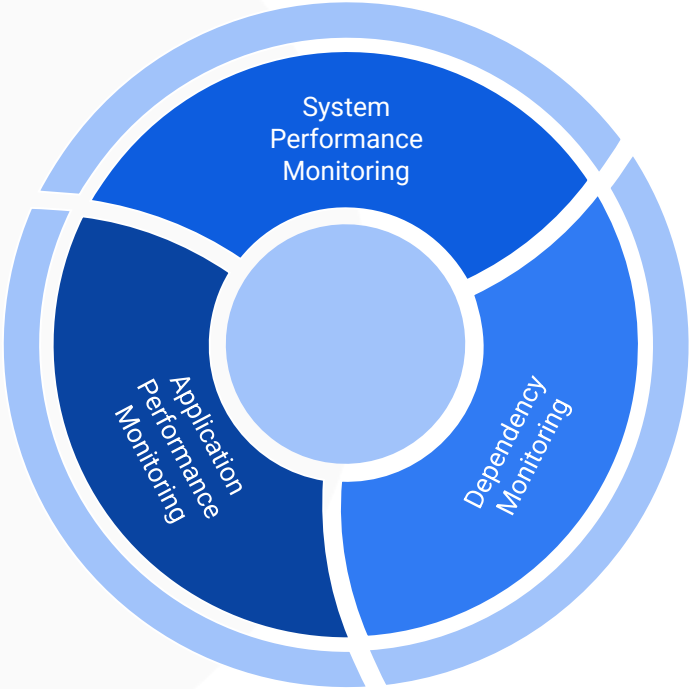


TIPS?

a) Be realistic; not every toil is bad and certainly not every toil can be ‘eliminated’
b) Set realistic timelines to reduce toil; once a quarter with monthly checkpoints is OK!
c) Set aside a few hours every week/month - e.g. Friday Hackathons for focused efforts
d) Invite experts from other teams to conduct Dojos and get a headstart (and sponsor lunch!)
e) Add more people to the team, maybe not permanently, but to enable breather
f) Reduce service expectations, e.g. lower the SLO!

WHY Lack of Monitoring Leads to Unintentional Chaos?

- When something breaks in production, teams need to be alerted to take action (preferably before the damage)
- For an alert to be fired, the system (app or infrastructure) needs to be be monitored
- That's where a 'good' monitoring system comes into play
- Tune your monitoring to be SLO-focused
- Monitor the "dependencies"
    - Probe downstream and generate trending data
    - Define SLOs considering the generated trending data



How does SRE help, again?

3 Themes – 1/3

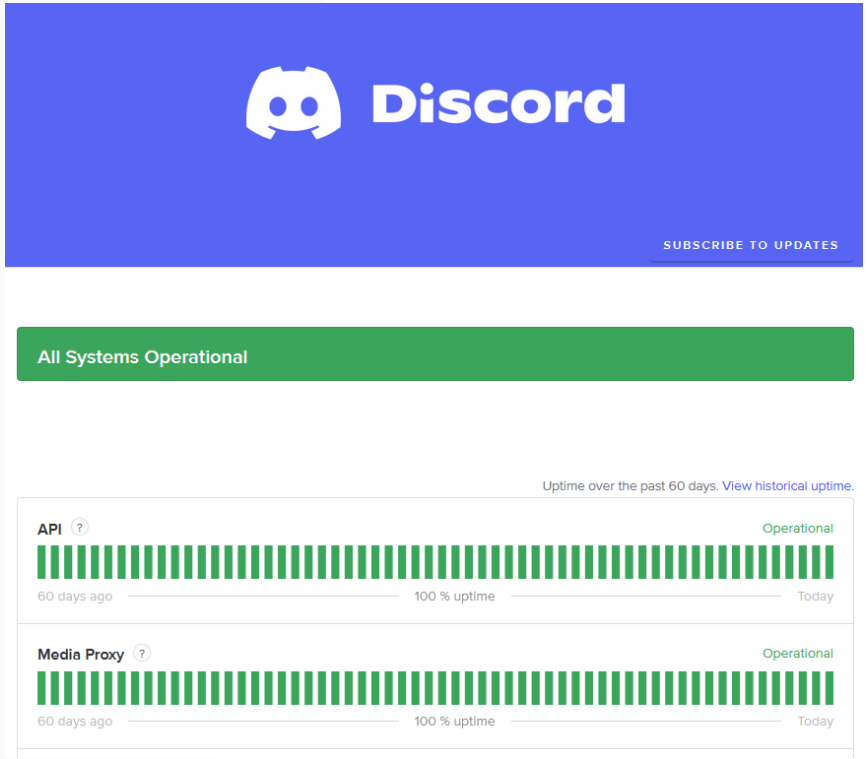
Speed

Number of deployments = as allowed by the SLO

3 Themes – 2/3

Vendor / Upstream / Downstream Dependencies

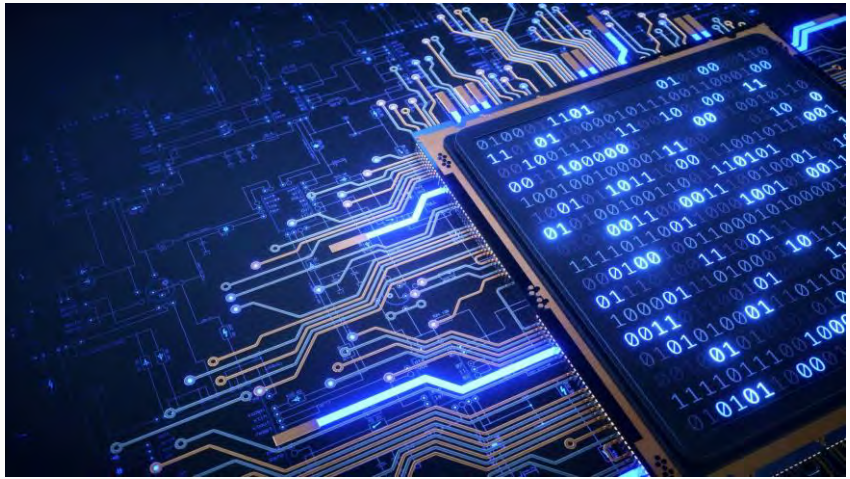
Go with data!



3 Themes – 3/3

Measuring ~~Everything~~

By tuning monitoring systems to SLOs and identifying SLO requirements for NEW features upfront, we can ensure a high signal-to-noise ratio!



Transformed!

Thank You!

Mandeep Ubhi

Email: mandeepsu@devsre.org
LinkedIn: mandeepsu