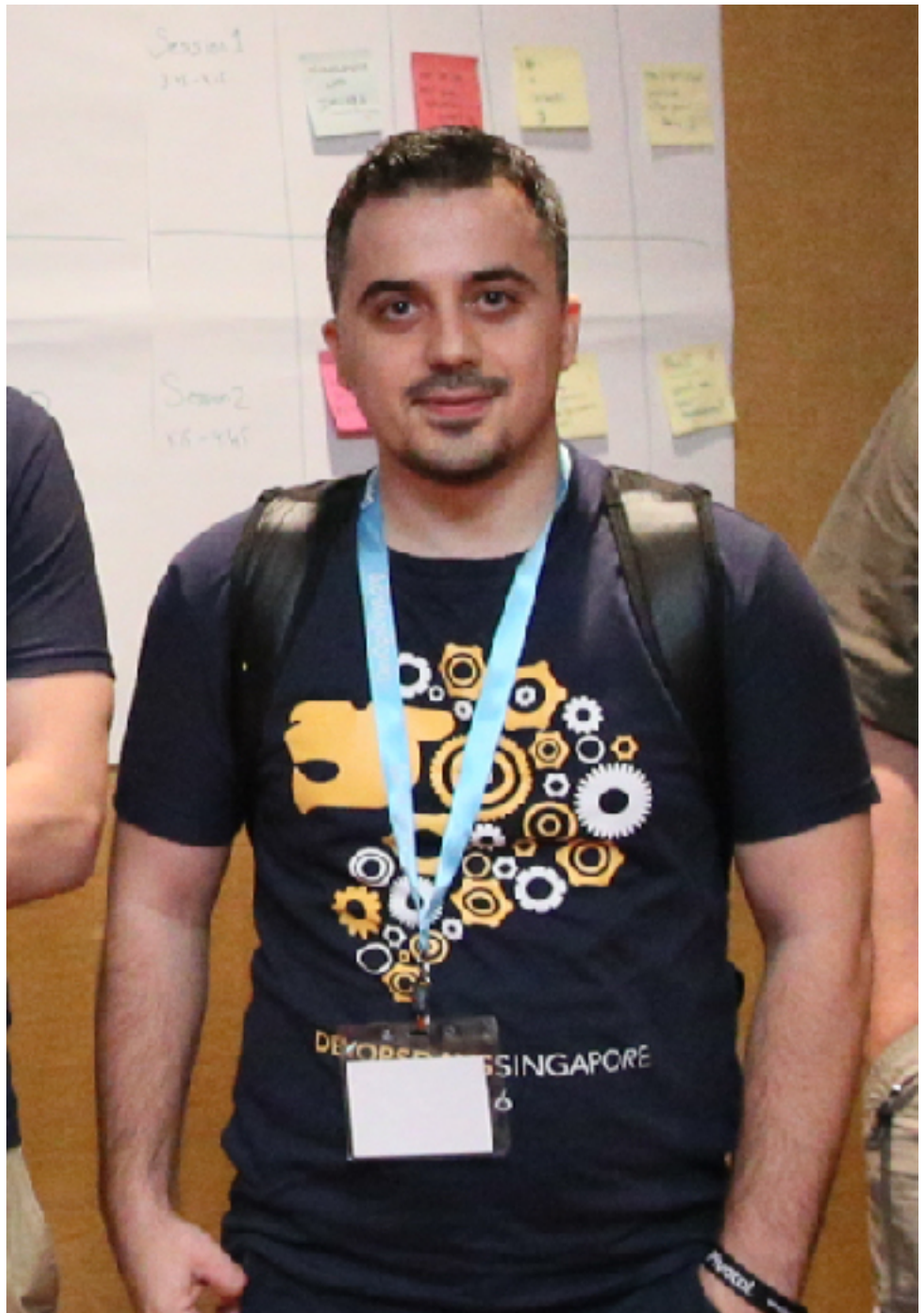
From Resilient to Antifragile Chaos Engineering Primer

By @Sergiu_Bodiu
Solution Architect

From Resilient to Antifragile Chaos Engineering Primer

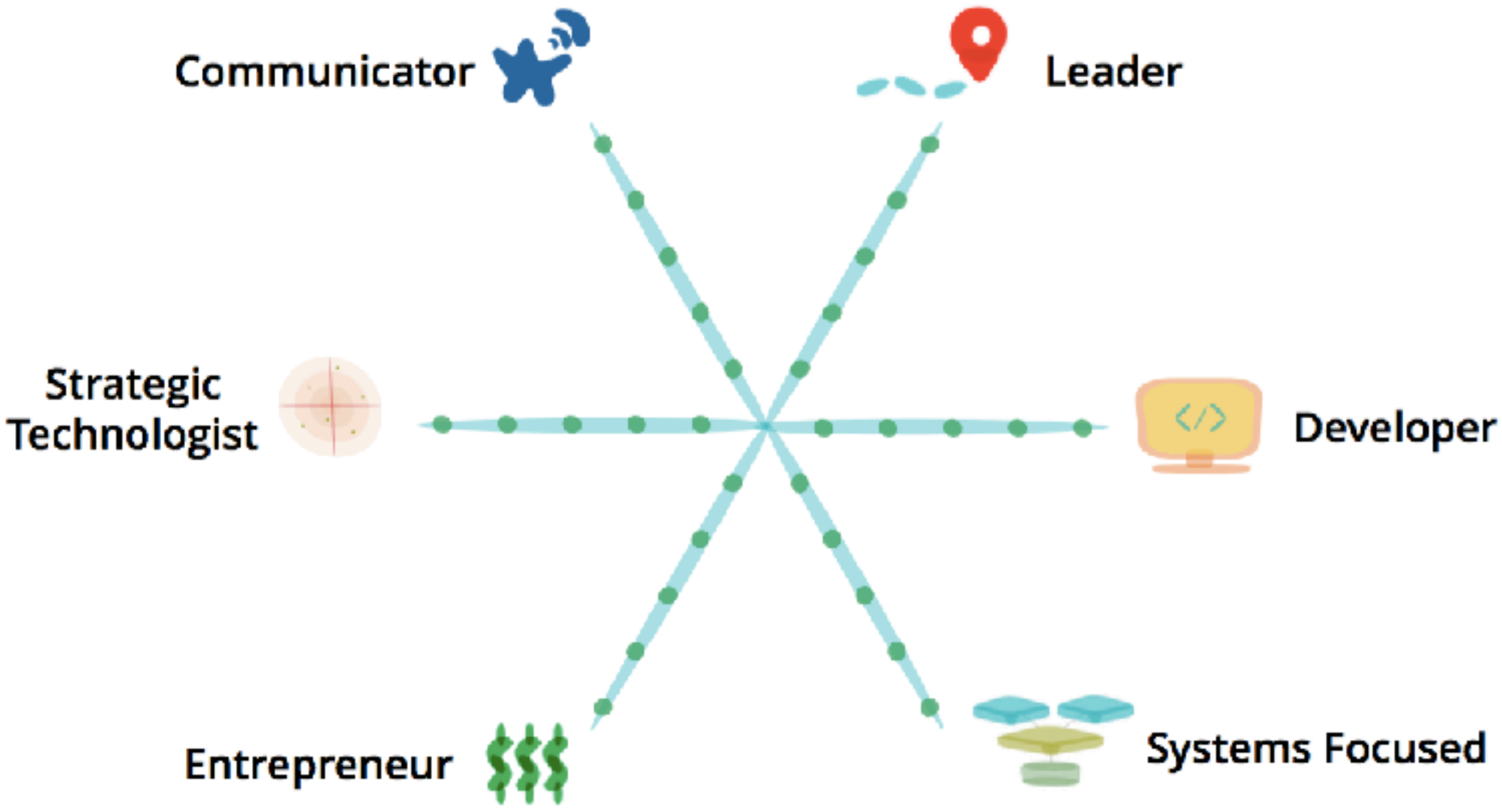
By @Sergiu_Bodiu
Solution Architect



Singapore Spring User Group

DevOpsDays Singapore Conference

what is an ARCHITECT



@patkua https://www.thekua.com/atwork/2016/11/the-well-rounded-architect/

Risk management

The new normal:

from RESILIENT

to **ANTIFRAGILE**

A new way to look at organizations

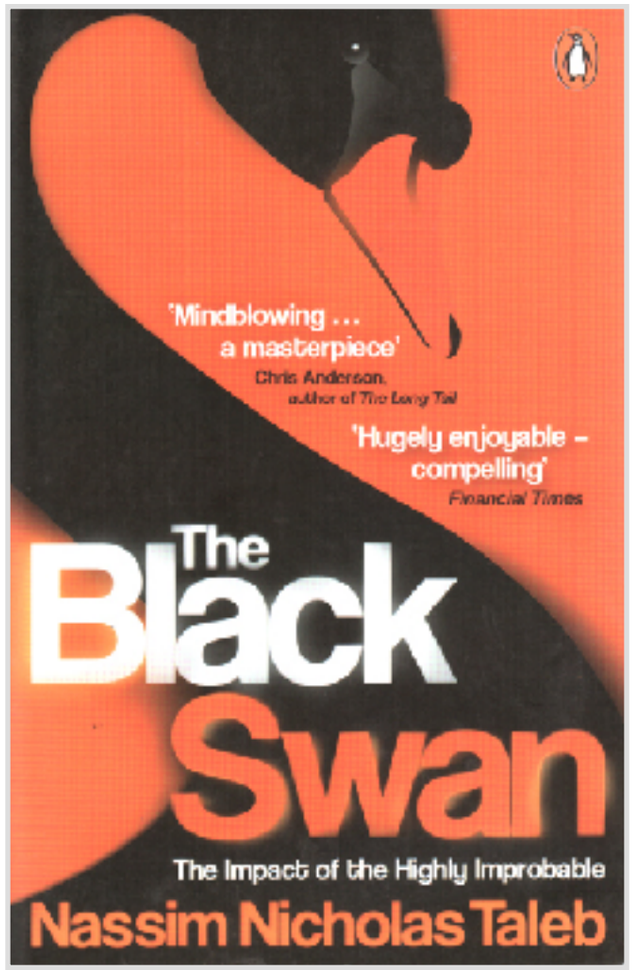
Fragile: **At risk of total failure / financial ruin**

Resilient: **Takes damage, avoids total failure, recovers**

Robust: **Absorbs uncertainty, repels blows, avoids damage**

Antifragile: **Responds to stress by mutating, maintains fitness for purpose. Identity Change.**

Blueprint for living in a Black Swan world.



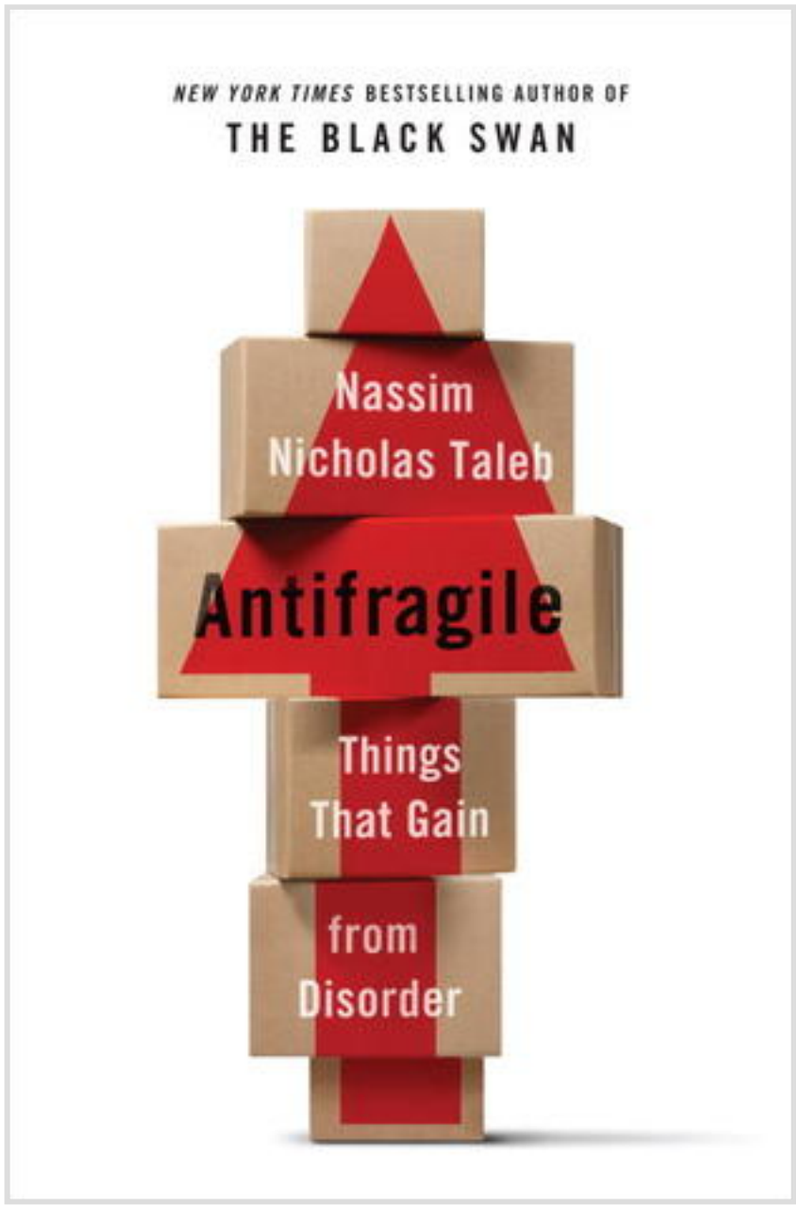
Antifragile

and only

the

Antifragile,

will Make it.

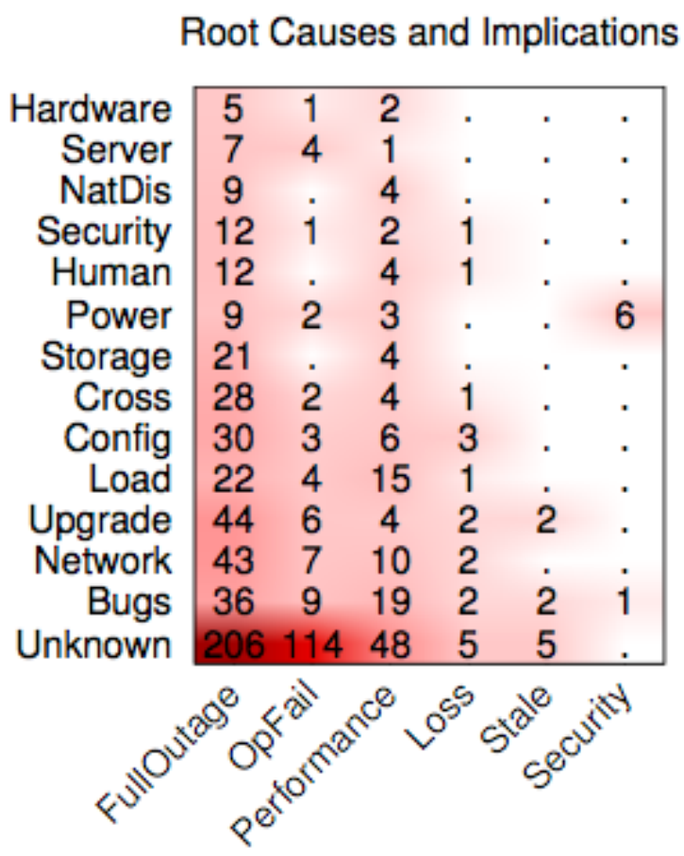


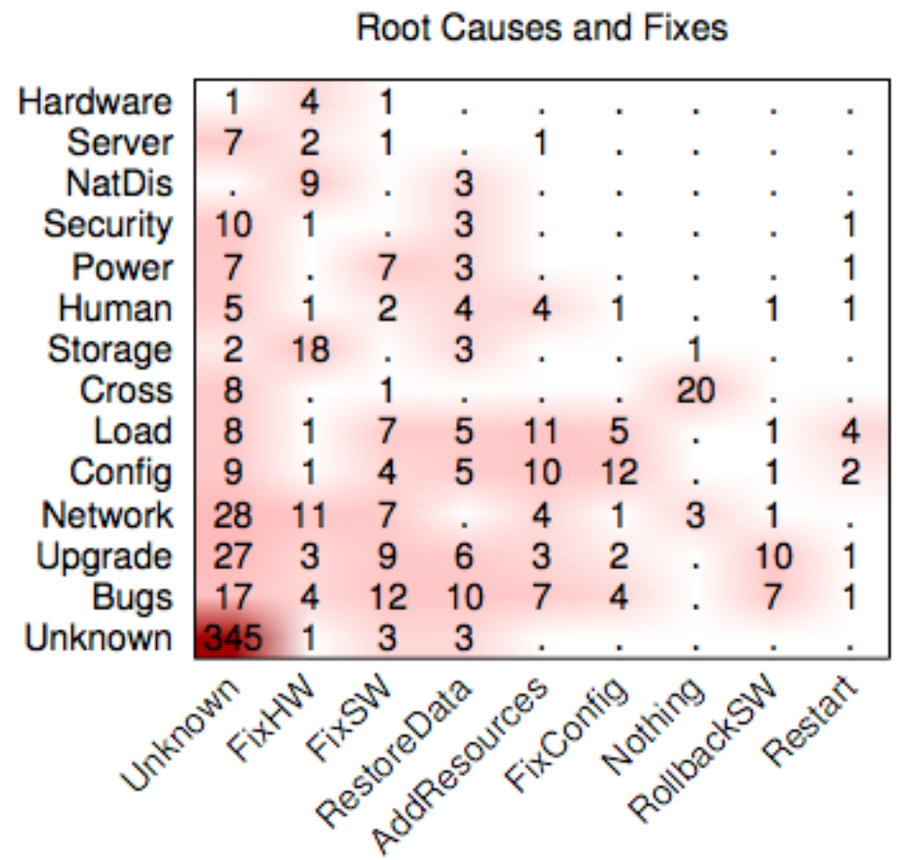
https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

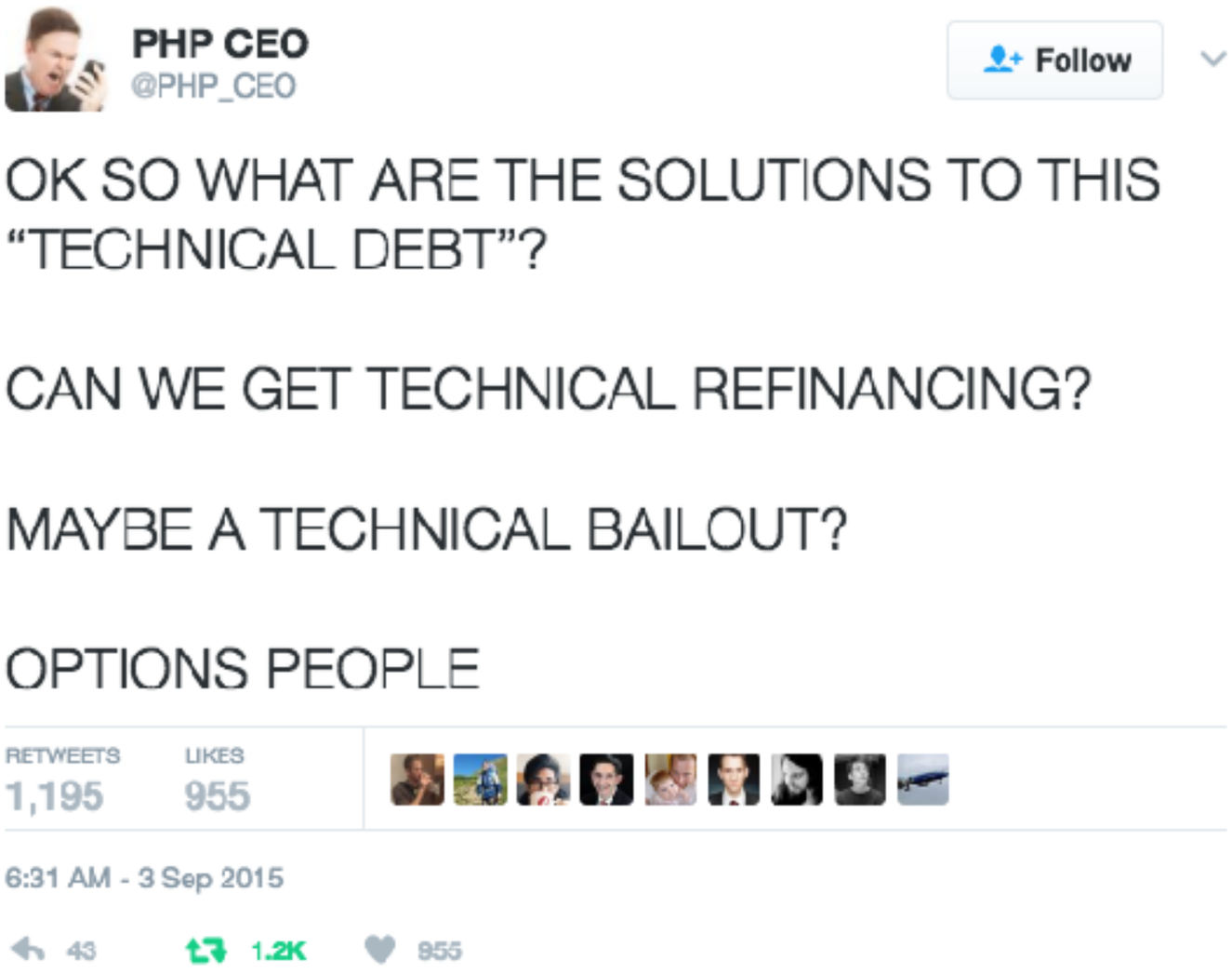
Software is Single Point of Failure

Root Cause Analysis: While component failures such as NETWORK, STORAGE, SERVER, HARDWARE, and POWER failures are anticipated and thus guarded with extra redundancies.





Distributed Systems Complexity



Complexity is like **Addiction**…

Case study: How complexity creeps in

- @jasonfried

https://m.signalvnoise.com/case-study-how-complexity-creeps-in-cba48023e6a1

NETFLIX

Cannot connect to the Netflix Service

Please Try Again

Chaos Engineering

Discipline of **experimenting** on a distributed system in order to **build confidence** in the system's capability to withstand turbulent conditions in **production**.

NETFLIX

Some outages in the Region

SingTel fined a record $6m for Bukit Panjang exchange fire;

Telstra goes down again, people can't drink beer or catch Ubers

Amazon Web Services outage causes Australian website chaos

Backups

"Backups always succeed.
It's the **restores that fail**.
Test your backups by practicing restores!"

Netflix Simian Army

Suite of tools for keeping your cloud operating in **top form**.

https://github.com/Netflix/SimianArmy

Chaos Monkey

1. Active during normal working hours
2. Break things in production
3. Design better software services
4. Embracing failure

http://techblog.netflix.com/2016/10/netflix-chaos-monkey-upgraded.html

https://github.com/Netflix/security_monkey

Monitor AWS and GCP accounts for policy changes and alerts on insecure configurations.

Security Monkey can be extended with custom account types, custom watchers, custom auditors, and custom alerters.

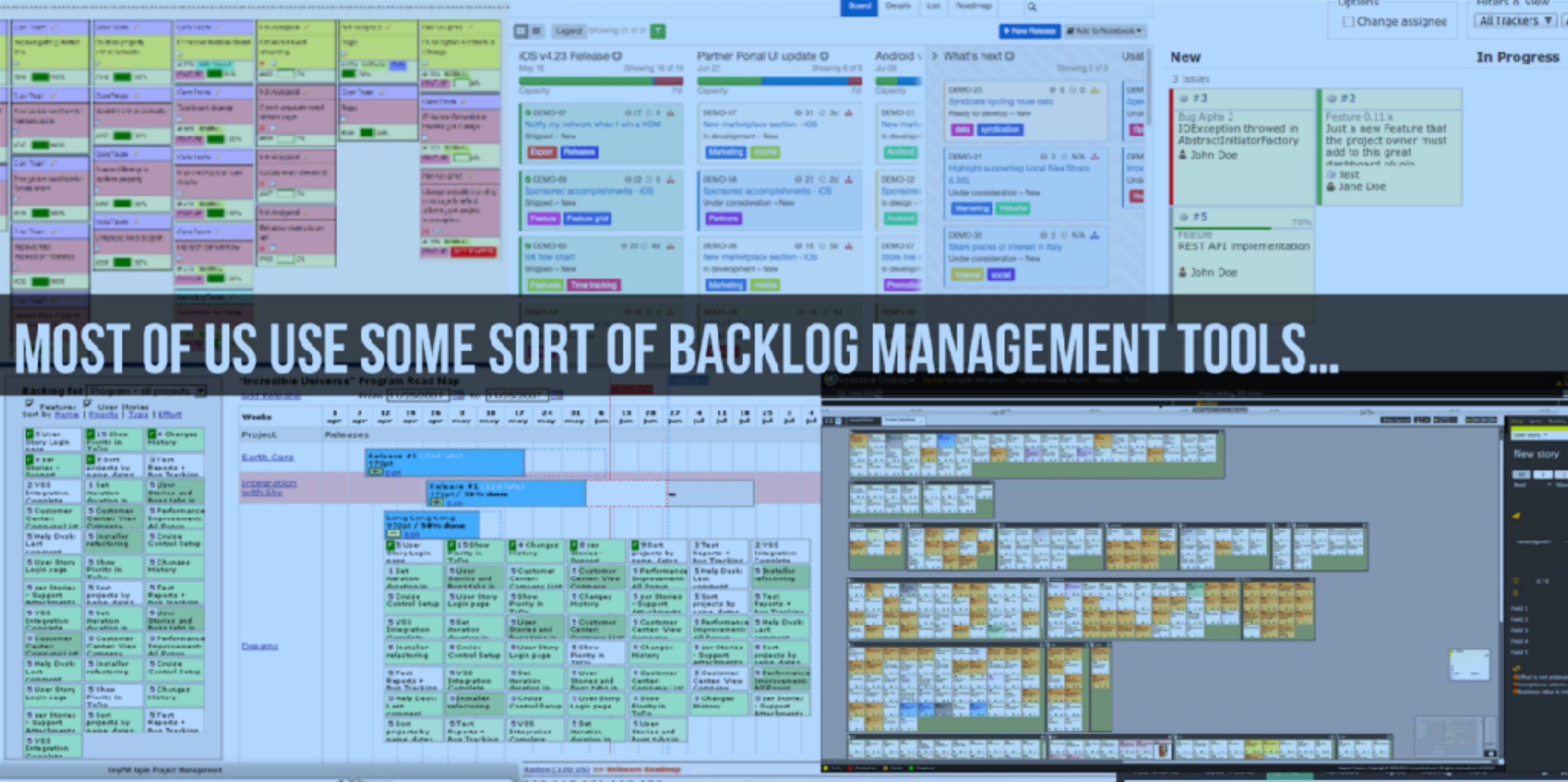
Other Monkeys

- Latency Monkey
- Janitor Monkey
- Conformity Monkey
- Doctor Monkey

PRINCIPLES > TOOLS

Why do we do >

What we do



MOST OF US USE SOME SORT OF BACKLOG MANAGEMENT TOOLS...

Principles of Chaos

1. Build a Hypothesis around Steady State Behavior
2. Vary Real-world Events
3. Run Experiments in Production
4. Automate Experiments to Run Continuously

Chaos Engineering Whitepaper 2016

Hypothesize

> sudo watch

- Start with steady state behavior.
- Monitor metrics that are visible
- Capture an interaction between the users and the system.

TIP: Utilisation is Virtually Useless as a Metric!

Vary Events

> sudo halt

- Terminate virtual machine instances
- Inject latency into requests between services
- Fail requests between services
- Fail an internal microservice
- Make an entire region unavailable

TIP: Select only a subset of users

Experiment

- End to end TESTING (Expensive)
    - Process is slow
    - Configuration Drfit from Production
- 92% ERRORS could be prevented (Simple)

TIP: Customers don't behave as your JMeter script.

https://www.usenix.org/system/files/conference/osdi14/osdi14paperyuan.pdf

Automate

> sudo while (1)

- Distributed systems changes continuously over time.
- Engineers modify the behavior of existing services, add new services.
- Engineers are changing runtime configuration parameters, upgrading and patching systems

TIP: Depending on the context, change the rate of each experiment.

Principles of Chaos

1. Build a Hypothesis around Steady State Behavior
2. Vary Real-world Events
3. Run Experiments in Production
4. Automate Experiments to Run Continuously

TIP: Intentionally break things, compare measured with expected impact, and correct any problems uncovered this way.

Chaos Engineering Whitepaper 2016

Reference Architecture for Cloud Native Platform

Infrastructure	Operations	Deployment	Runtime & Data	Security
Container Orchestration	Service Monitoring and Dependency Management	Lifecycle Management Deploy | Patch | Ugrade | Retire	HTTP / Reverse Proxy	Control Plane Audit & Compliance
Infrastructure Orchestration	Inventory, Capacity, and Management	Release Packaging, Management & Deployment	Application Runtime	Security Event & Incident Management
Service Discovery	Event Management and Routing	CI Orchestration	In-Memory Object Cache	Secrets Management
Configuration Management	Persistent Team Chat	TDD Frameworks	Search	Certificate Management
Core IaaS	Metrics & Logging Analytics & Visualization	Artifact Repository	Messaging	Identity Management
NAT / DNS	Log Aggregation, Indexing & Search	Standard Builds & Configurations	NoSQL Docuemnt Store	Threat & Vulnerability Scanning
SDN / IPAM	Metrics Collection, Storage & Retrieval	Source Control Management	NoSQL Key/Value Store	Network Security
Firewalls / WAN & VPN				
Storage / Load Balancers				
Compute / Network				

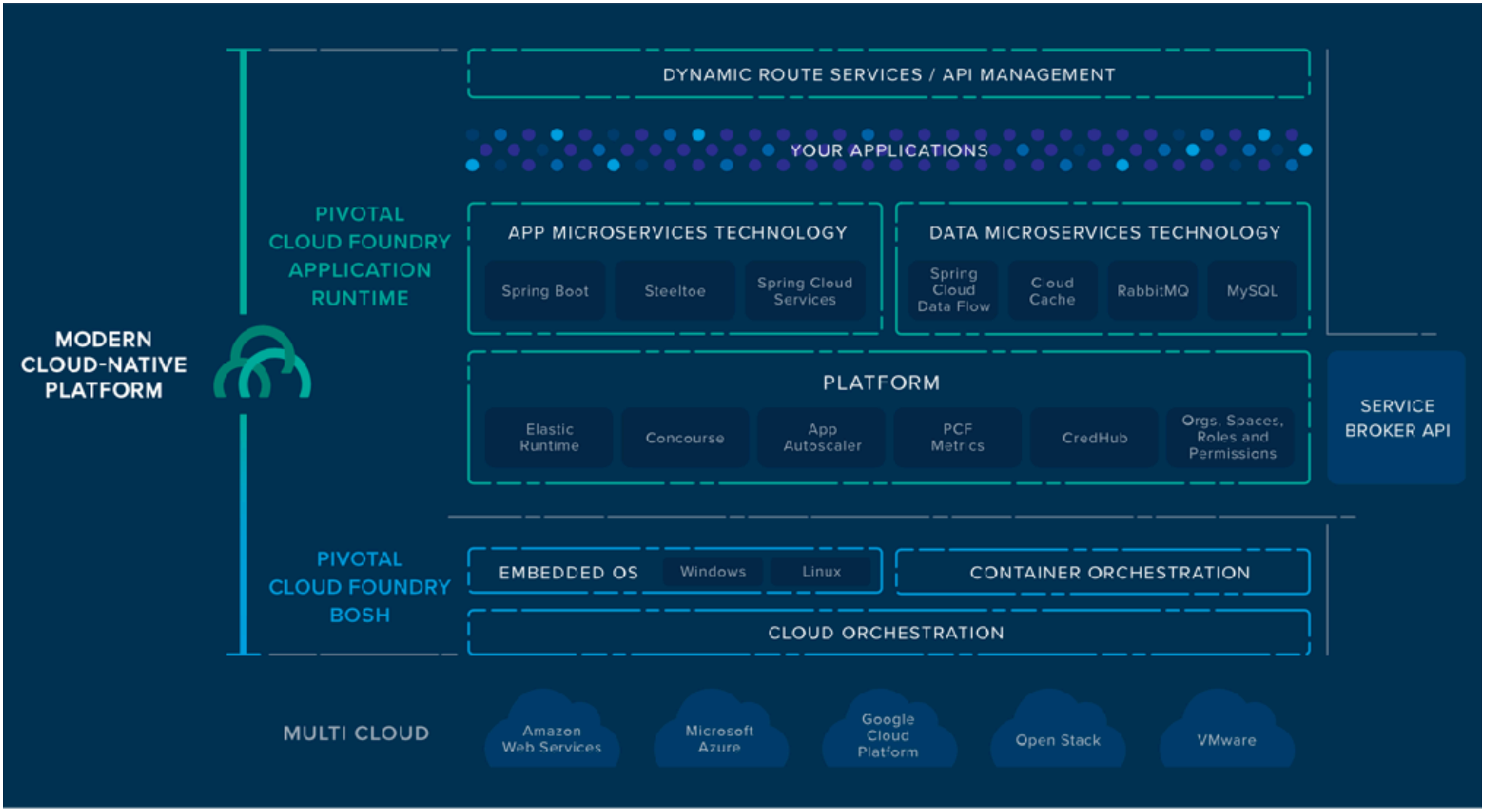
Figure 2 - The 5 domains of a cloud-native platform.

Pivotal

https://content.pivotal.io/white-papers/the-upside-down-economics-of-building-your-own-platform

Pivotal Cloud Foundry

Pivotal



Chaos Lemur demo

Chaos Lemur =

Chaos Monkey + PCF

Locust demo

Locust is an open-source Python load testing framework.

- Define user behaviour in code
- Can execute end-to-end user test with sessions and cookies.
- Expands to multiple slaves to increase load capacity
- Allows for distributed user paths based on percentages

Gatling is an open-source Scala load testing framework

- High performance
- Ready-to-present HTML reports
- Scenario recorder and developer-friendly DSL

Lessons Learned

- **Systematic approach** to Chaos Testing
  - This is incredible hard under pressure.
- **Don’t wait** so long to start load testing.
  - The conversations drive new requirements.
  - Changing architecture last minute is extremely dangerous.
- **Join** the community
  - Build relation with Networking Team, Database Team, Third Party Partners, Vendors etc..
  - Make everything Asynchronous (Embrace Failure, Background Tasks, Retry, Idempotence)

The importance of reliability

Don't trust claims systems make about themselves & their dependencies.

Verify by breaking.

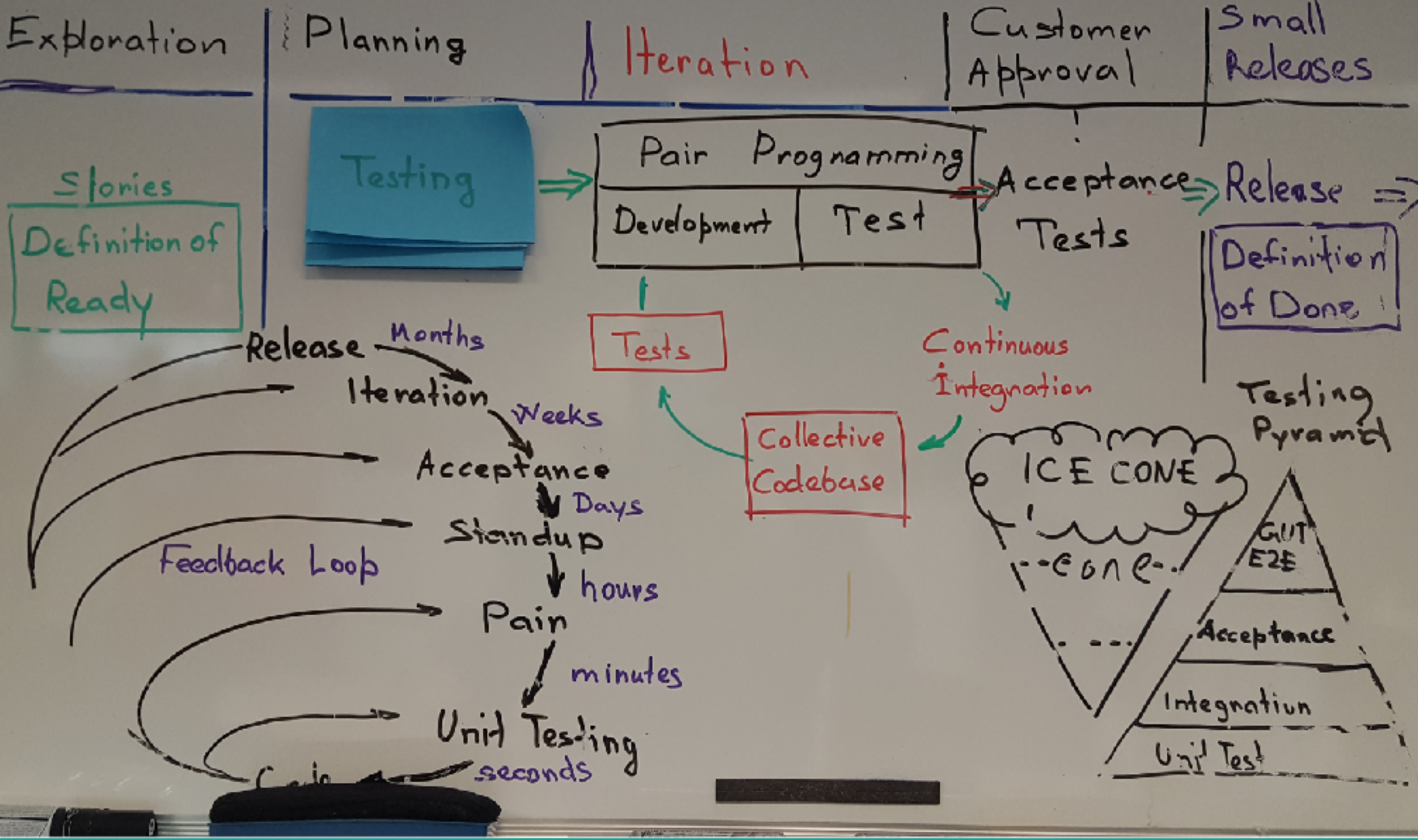
Clean your process

Culture > Principles > Tools

> sudo halt
Incident Start
Impact

> Post Mortem

Testing Pyramid



https://watirmelon.blog/2012/01/31/introducing-the-software-testing-ice-cream-cone/

Further Reading

https://www.infoq.com/br/presentations/exercising-failure-at-netflix
https://www.infoq.com/podcasts/failure-as-a-service
https://www.infoq.com/articles/chaos-engineering

@Ops_Engineering https://www.youtube.com/watch?v=CZ3wIuvmHeM
@caseyrosenthal https://www.youtube.com/watch?v=Q4nniyAarbs

Peter Alvaro: Orchestrated Chaos: Applying Failure Testing Research at Scale
Adrian Colyer Simple Testing Can Prevent Most Critical Failures

Questions

Thank You

@sergiu_bodiu

Principles

Any developer building applications which run as a service. Ops engineers who deploy or manage such applications.

https://12factor.net:

Anyone working in software that writes tests or maintains continuous integration pipelines.

http://www.10factor.ci