How many **9s** are enough?

Kolton Andrus, CEO

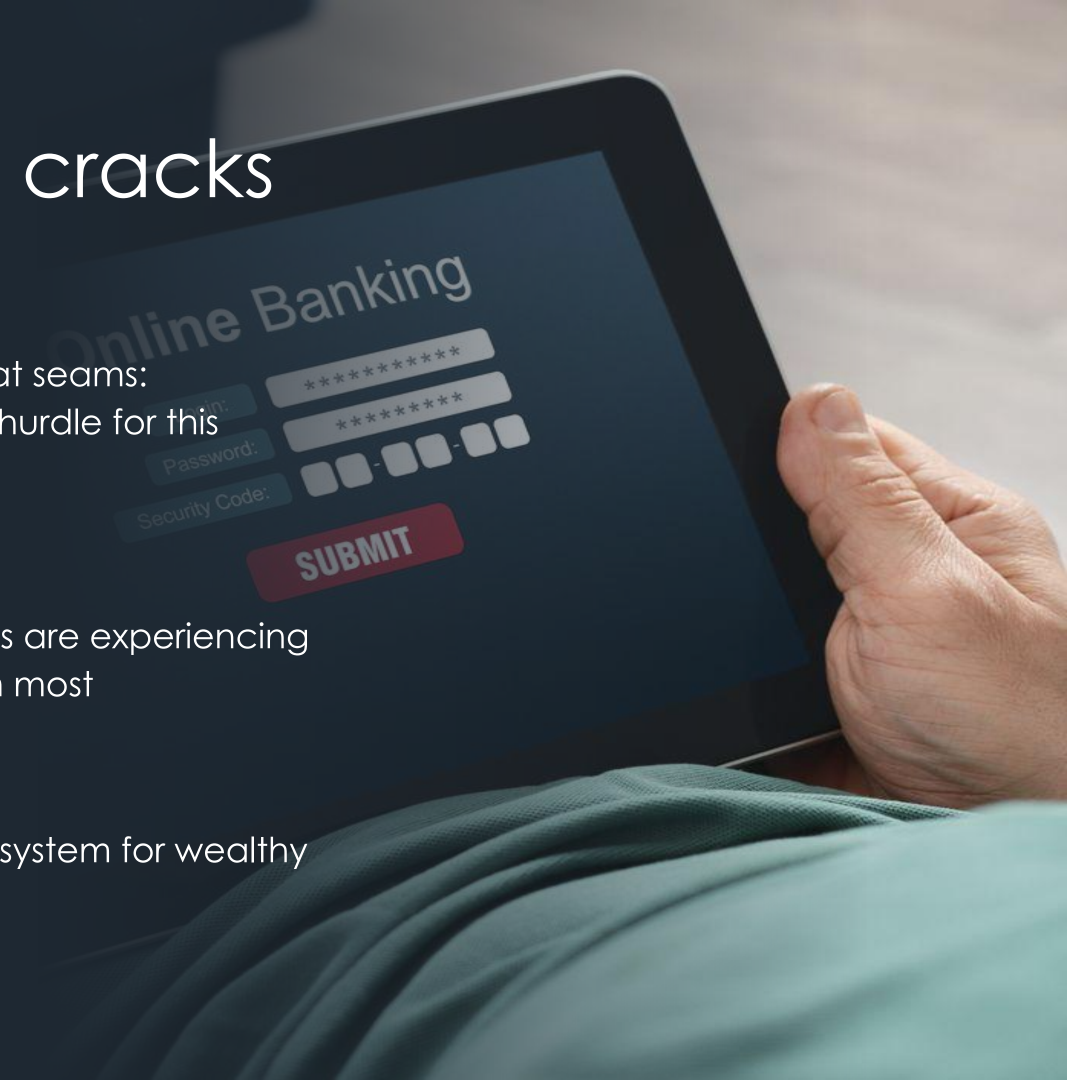
Gremlin

Demand revealing cracks

Zoom goes boom, Teams tears at seams: Technology stumbles at the first hurdle for this homeworking malarkey
3.16.20

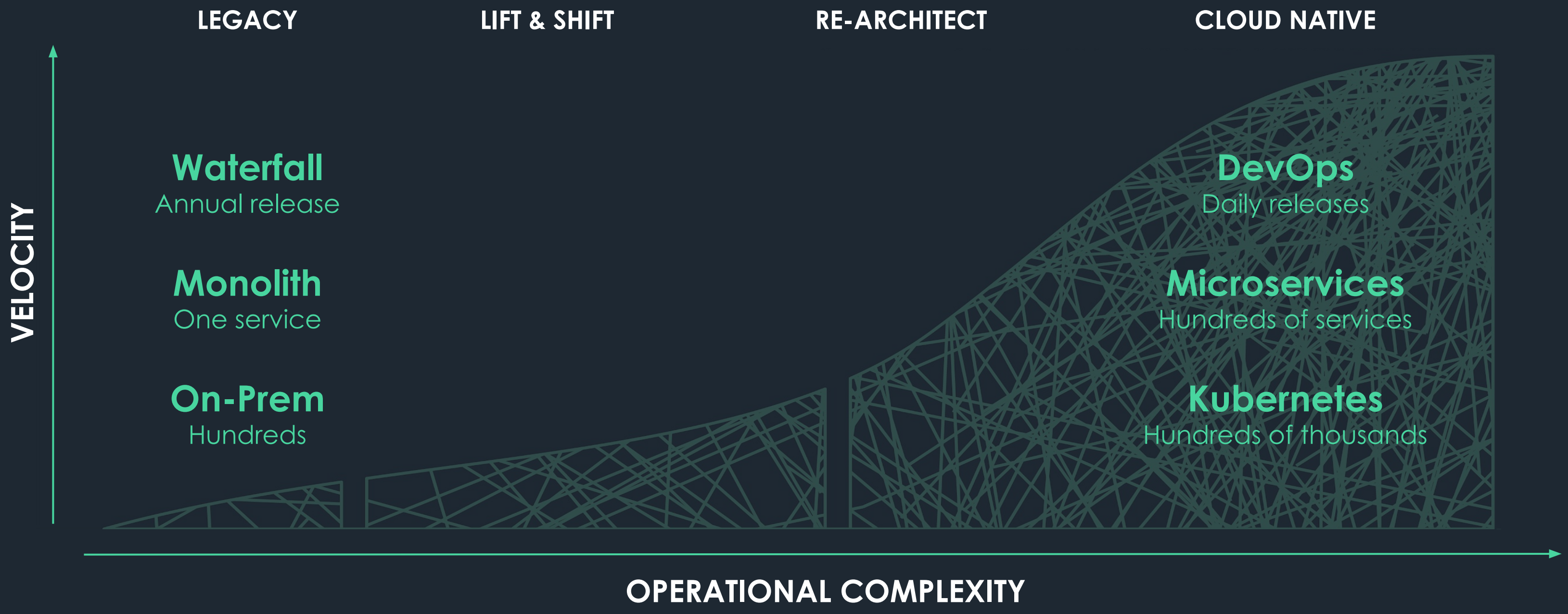
Nintendo’s online Switch services are experiencing an outage when we need them most
3.17.20

Morgan Stanley's online trading system for wealthy clients went down
3.25.20



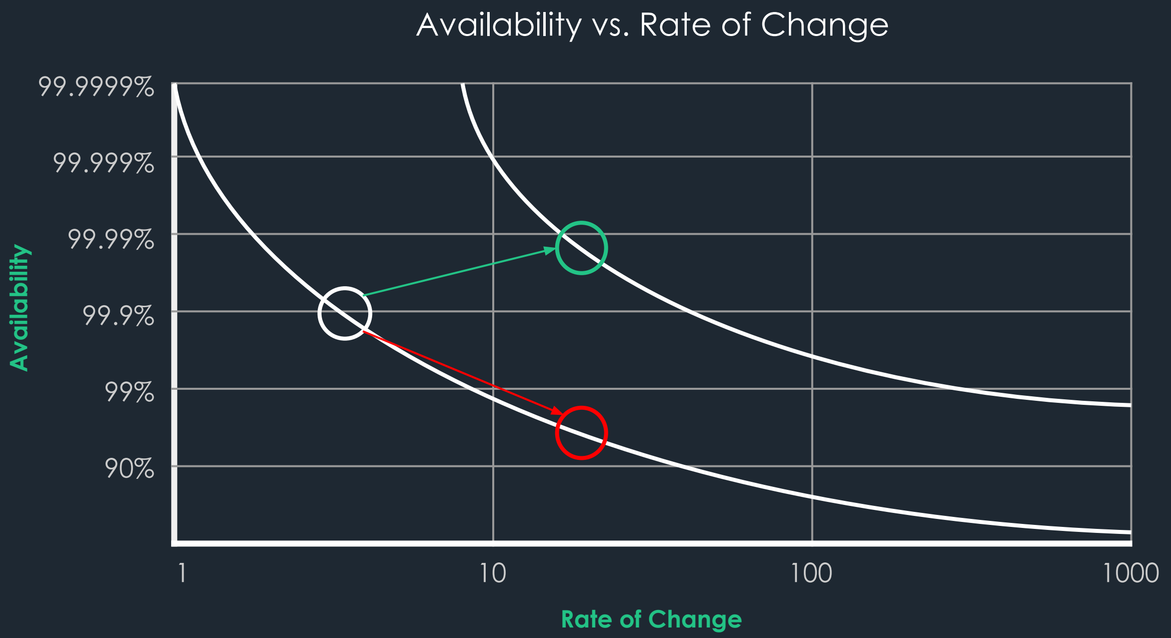
Velocity Comes at a Price

Modern applications are exponentially more complex



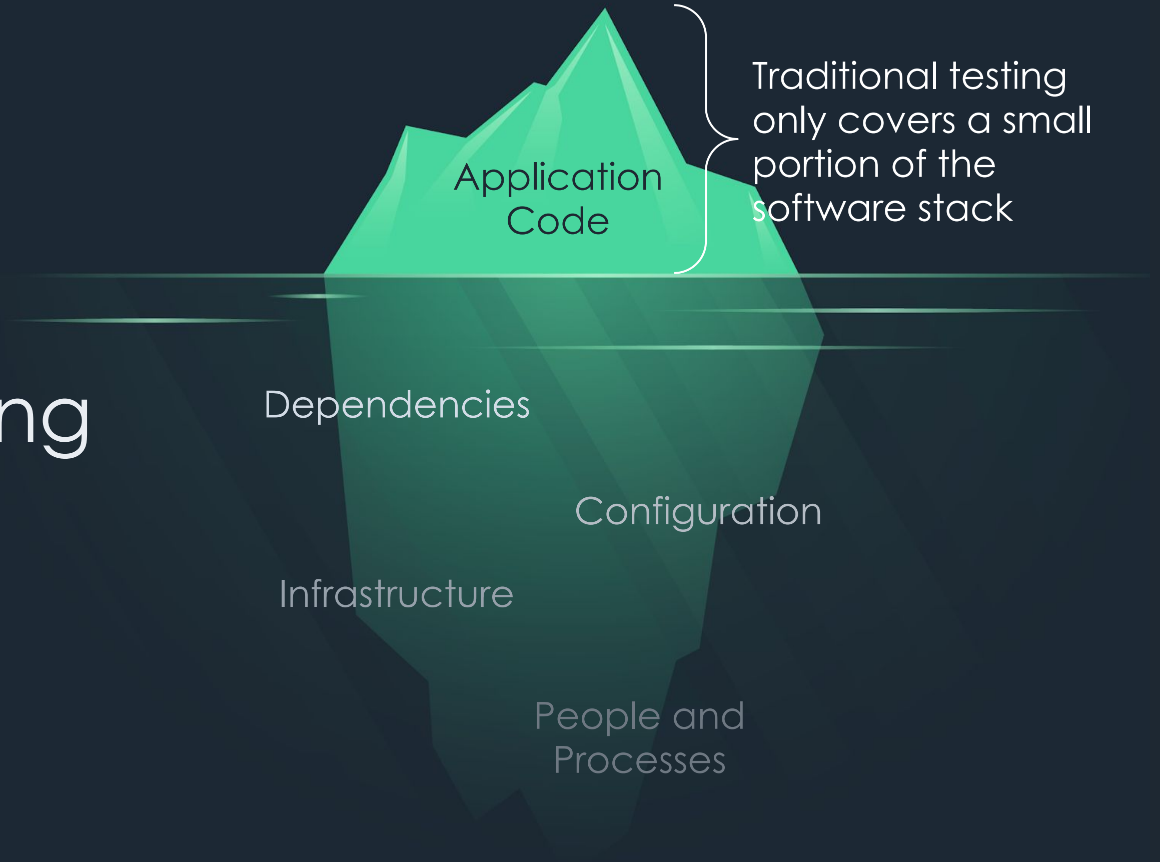
Gremlin

Invest in velocity *and* reliability



> *Change introduces new forms of failure that are difficult to see before the fact....*
>
> – Richard Cook, How Complex Systems Fail

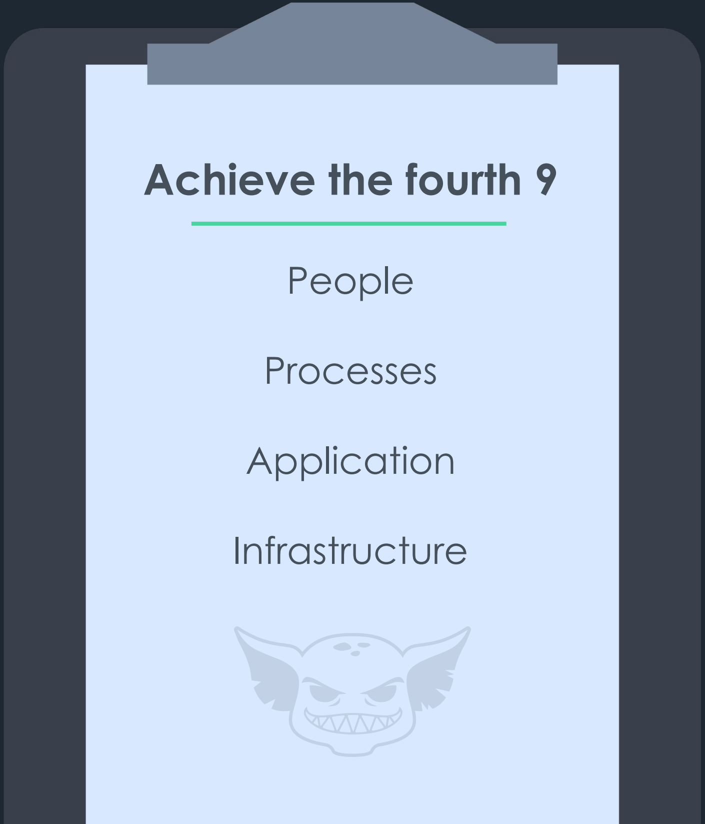
Traditional Testing **is not enough**.



Traditional testing only covers a small portion of the software stack

Chaos Engineering

Thoughtful, controlled experiments designed to reveal the weakness in our systems.



Achieve the fourth 9

People

Processes

Application

Infrastructure

Outages **101**

Anatomy of an incident

NETFLIX

Starts per Second (SPS)

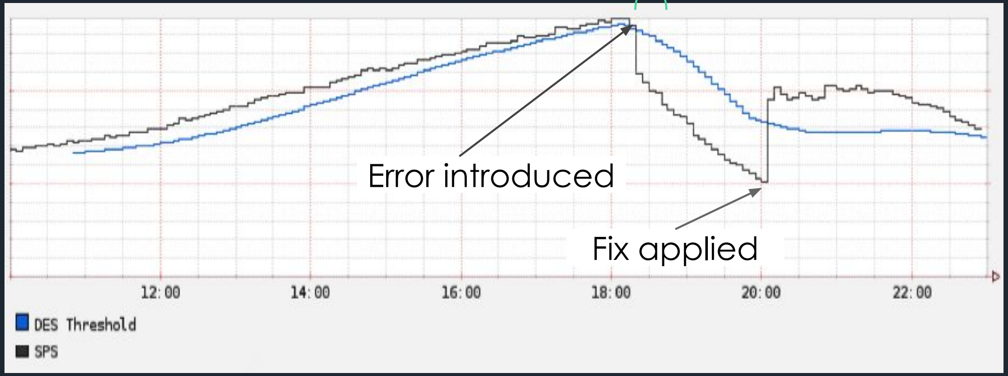
Key metrics to track

Mean time to detection (MTTD)

Mean time to resolution (MTTR)

Mean time between failures (MTBF)

Time to detection



Time to resolution

Pick a metric that is meaningful

- Start with monitoring user requests or using APM synthetic data
- Define your SLOs/SLAs
- Set a static threshold for acceptable failed requests
- Iterate and improve that threshold, remove noise
- Move to setting cyclical, seasonal, or moving thresholds

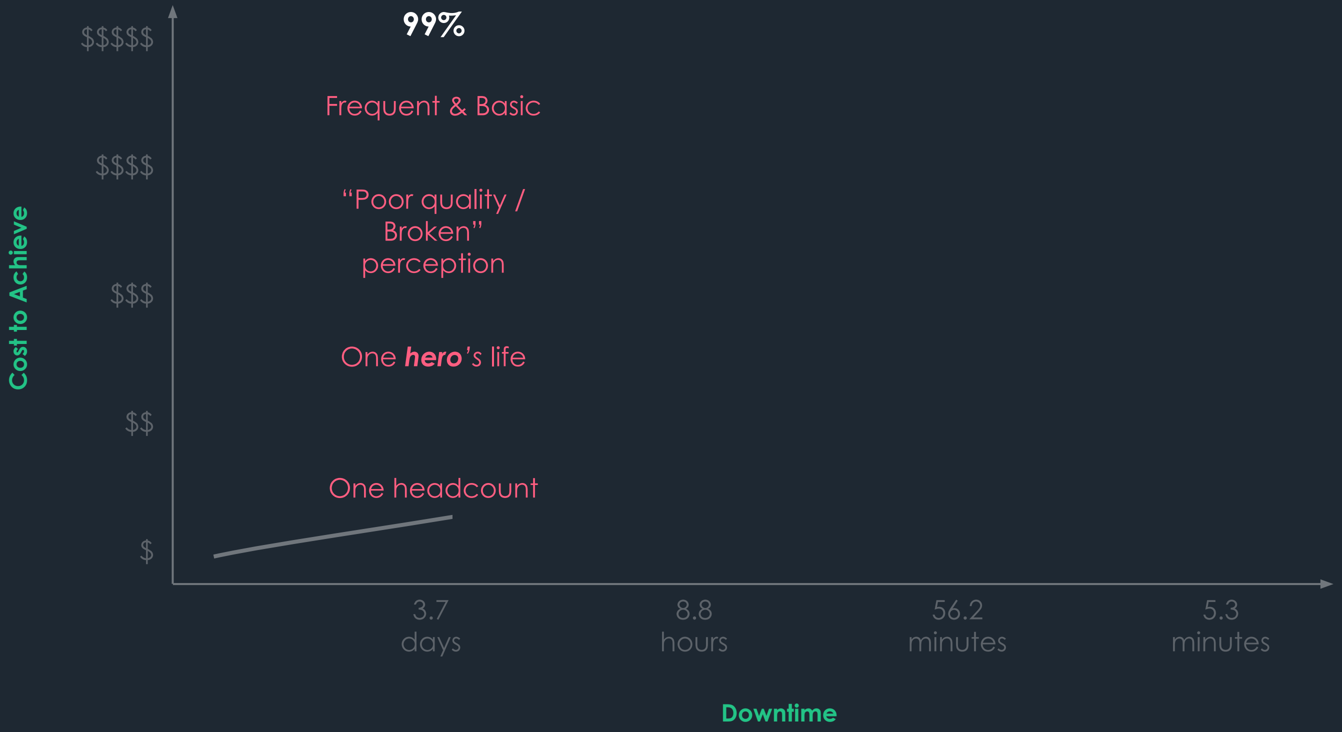
Call Leader **Basics**

- Have **one person** responsible for making decisions
- Report status **updates often**
- Have **participants join** the call and mute ("Who's typing in the background?")
- **Coordinate changes** so teams are not acting in isolation
- **Excuse people** when they are no longer needed
- Have a **single 'owner'** to drive the incident review and analysis

What’s the right **number of 9s** for you?

TLDR: Not everyone is Netflix

Two Nines: What the world looks like



	99%
Outages	Frequent & Basic
Customer Impact	“Poor quality / Broken” perception
Employee Impact	One ***hero***’s life
Learnings	One headcount

Two Nines:

"Brent"

99.999%

Two Nines:

The world today	What to improve
Basic logging; little to no monitoring	Add monitoring and alerting
Unit and integration tests	Automate build and deploy pipelines
Ad-hoc incident management process (AKA NONE)	Create an incident management program; validate by running a Fire Drill
Lack of Redundancy	Add capacity and zone redundancy, test zone failures

Fire drills prepare us to respond quickly, calmly, and safely.

Verify Monitoring with Chaos Engineering to **avoid missed alerts or prolonged outages**

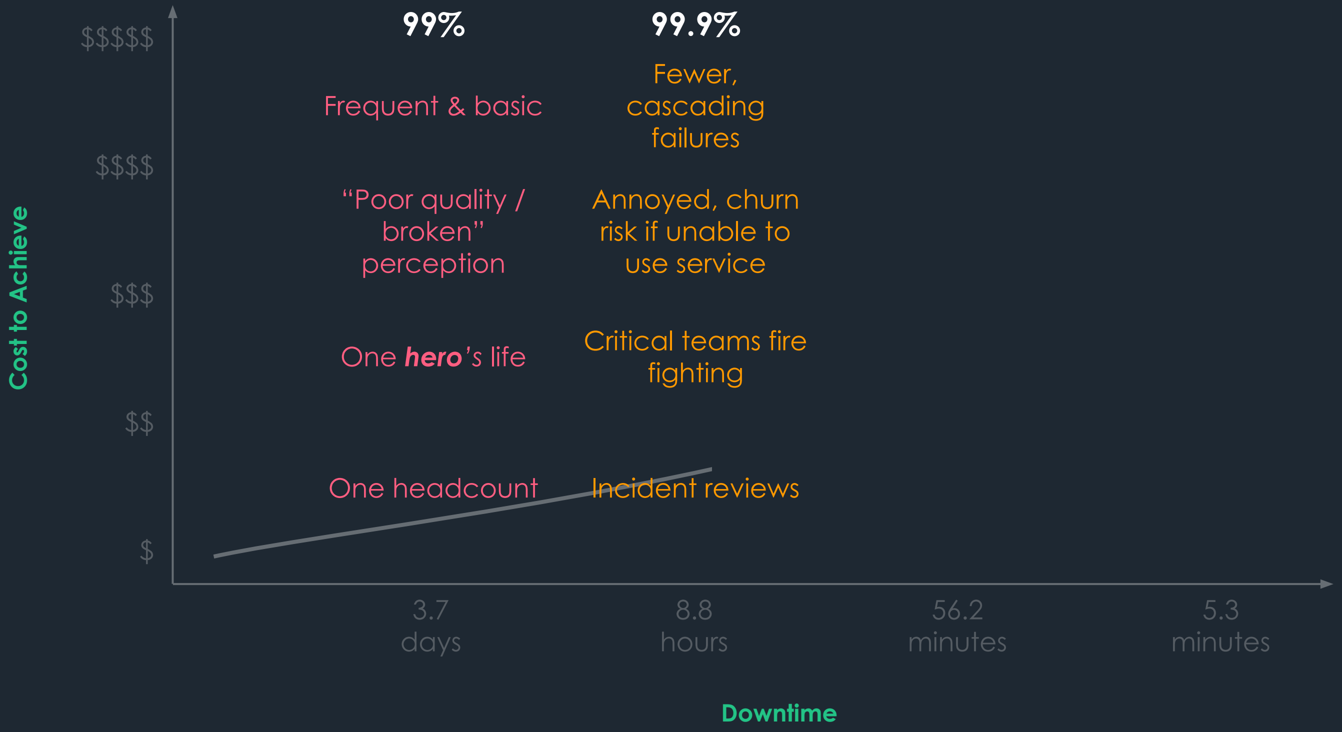
Experiments

- CPU spike on your service to simulate runaway processes
- Slow response from your database
- Recreate a past incident to compare your team's recovery time

Investment

- Weekly incident review meetings
- Monthly GameDays
- Quarterly Fire Drills

Three Nines: What the world looks like



	99%	99.9%
Outages	Frequent & basic	Fewer, cascading failures
Customer Impact	"Poor quality / broken" perception	Annoyed, churn risk if unable to use service
Employee Impact	One ***hero***'s life	Critical teams fire fighting
Learnings	One headcount	Incident reviews

99.999%

Three Nines:

SRE Team

99.999%

Gremlin

Three Nines:

The world today

Logging and Monitoring,
but may be noisy and scattered

Building and deploying; more frequent
code changes across teams

Incident reviews happening;
learnings may be isolated

Infrastructure is zone redundant

What to improve

Reduce noise by tuning thresholds

Add canary deploys and failure testing

Share learnings and best practices
across teams

Move to Regional redundancy;
test region failover

Prepare for dependency failure and **reduce the time to resolve** issues

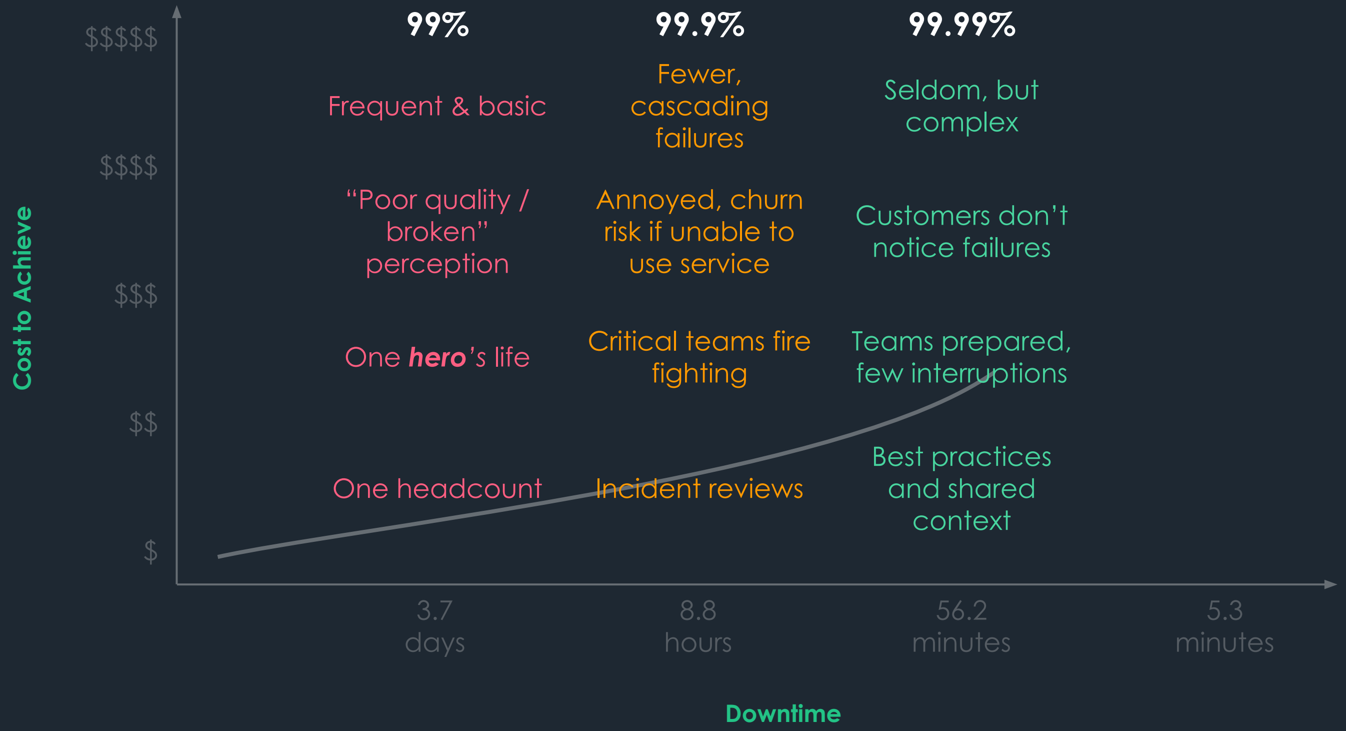
Experiments

- Simulate slow or lost network connectivity between nodes
- Test node, pod, zone failure
- Service unable to reach DNS

Investment

- Enact incident management process
- “Tier 1” services failure testing weekly
- Failure and load testing in build and deploy pipeline

Four Nines: What the world looks like



	99%	99.9%	99.99%
Outages	Frequent & basic	Fewer, cascading failures	Seldom, but complex
Customer Impact	"Poor quality / broken" perception	Annoyed, churn risk if unable to use service	Customers don't notice failures
Employee Impact	One ***hero***'s life	Critical teams fire fighting	Teams prepared, few interruptions
Learnings	One headcount	Incident reviews	Best practices and shared context

99.999%

Four Nines:

Culture of Resilience

99.999%

Gremlin

Four Nines:

The world today

Observability is ubiquitous

Unit, Integration, Performance and Failure Testing as part of pipeline

Company-wide GameDays, blameless post-mortems, frequent trainings

Region redundant, active-active architecture

What to improve

Anomaly detection and analysis

Canary and regional rollout, proactive exploration of potential failure modes

Practice to prevent atrophy through fire drills and mock events

Multiple infrastructure providers and redundant third party services

99.999%

Stress your cloud architecture to ensure it is **configured for reliability**

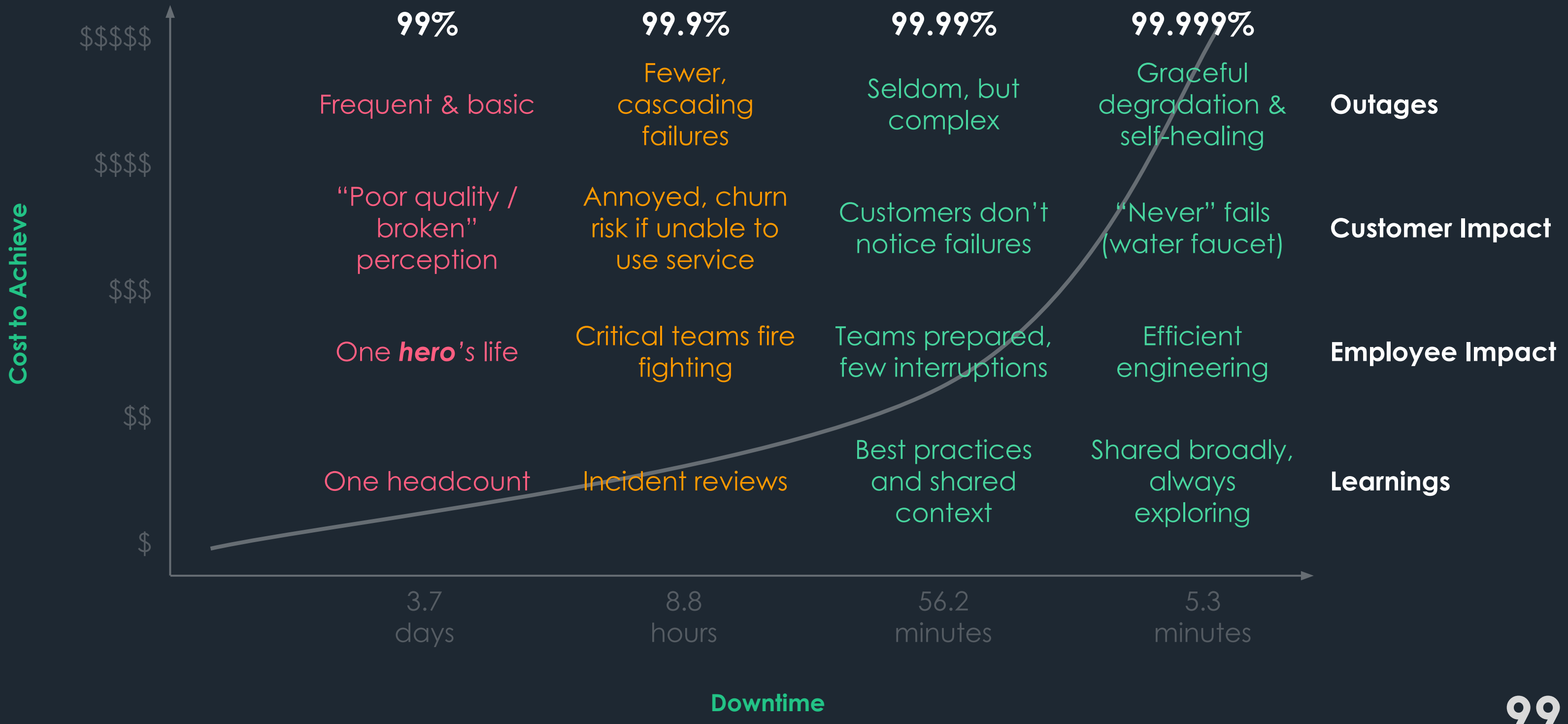
Experiments

- Handle degraded networks
- Peak traffic spikes
- DNS resolver failure
- Test region evacuation

Investment

- Bimonthly failover exercises
- All Services testing weekly
- Test in production

Five Nines: What the world looks like



	99%	99.9%	99.99%	99.999%
Outages	Frequent & basic	Fewer, cascading failures	Seldom, but complex	Graceful degradation & self-healing
Customer Impact	"Poor quality / broken" perception	Annoyed, churn risk if unable to use service	Customers don't notice failures	"Never" fails (water faucet)
Employee Impact	One ***hero***'s life	Critical teams fire fighting	Teams prepared, few interruptions	Efficient engineering
Learnings	One headcount	Incident reviews	Best practices and shared context	Shared broadly, always exploring
Downtime	3.7 days	8.8 hours	56.2 minutes	5.3 minutes

Five Nines:

The Future

99.999%

Five Nines:

You're Done!

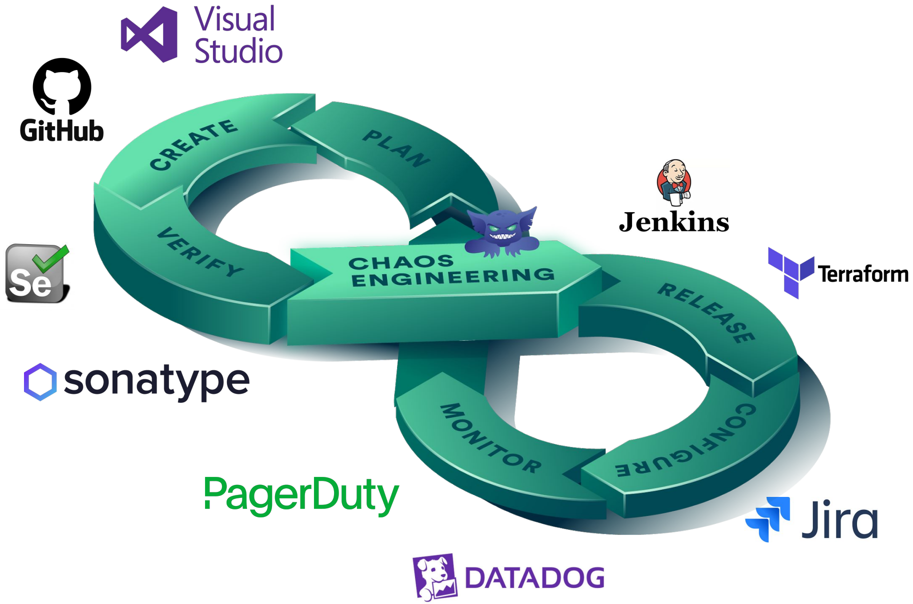
99.999%

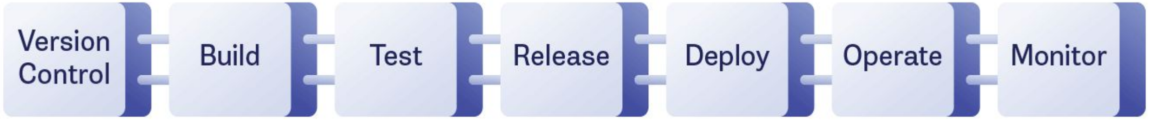
Thank you!

gremlin.com/community

Contain the Blast Radius

The modern engineering toolchain




Version Control

Build

Test

Release

Deploy

Operate

Monitor

Limit the Business Impact of COVID-19 Coronavirus Outbreaks by Improving Infrastructure Resilience

> “*Improving infrastructure resiliency can* ***protect organizations from significant business disruptions****… Organizations that can expect increased demand during a pandemic must be able to scale up capability to handle* ***exponential workload increases in a relatively short time.***”

– Gartner®