Before, During, and After Chaos

•••

Creating foresight through a cyclic approach

@nora_js

Before, During, and After Chaos

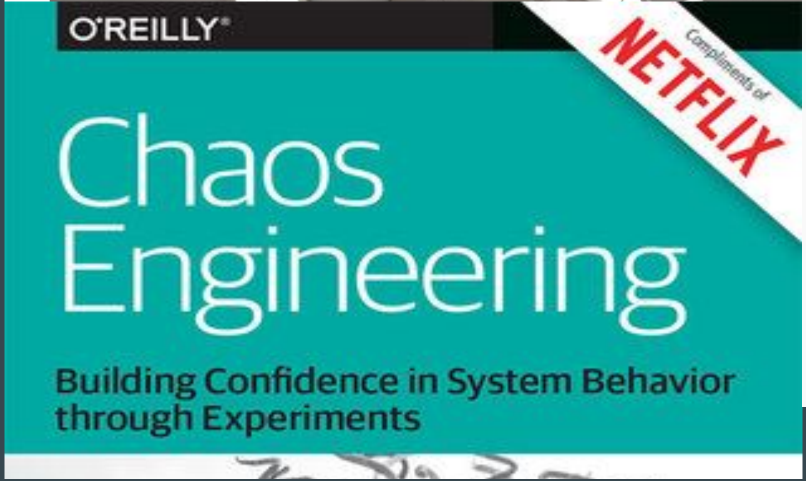
•••

Creating foresight through a cyclic approach

@nora_js

Human Factors & System Safety

FACULTY OF ENGINEERING, LTH

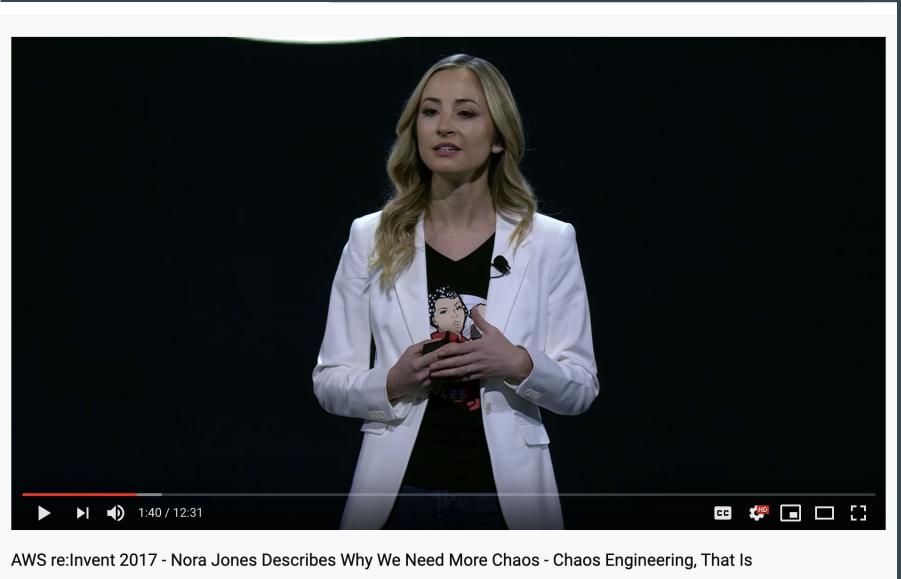


O'REILLY®

Compliments of NETFLIX

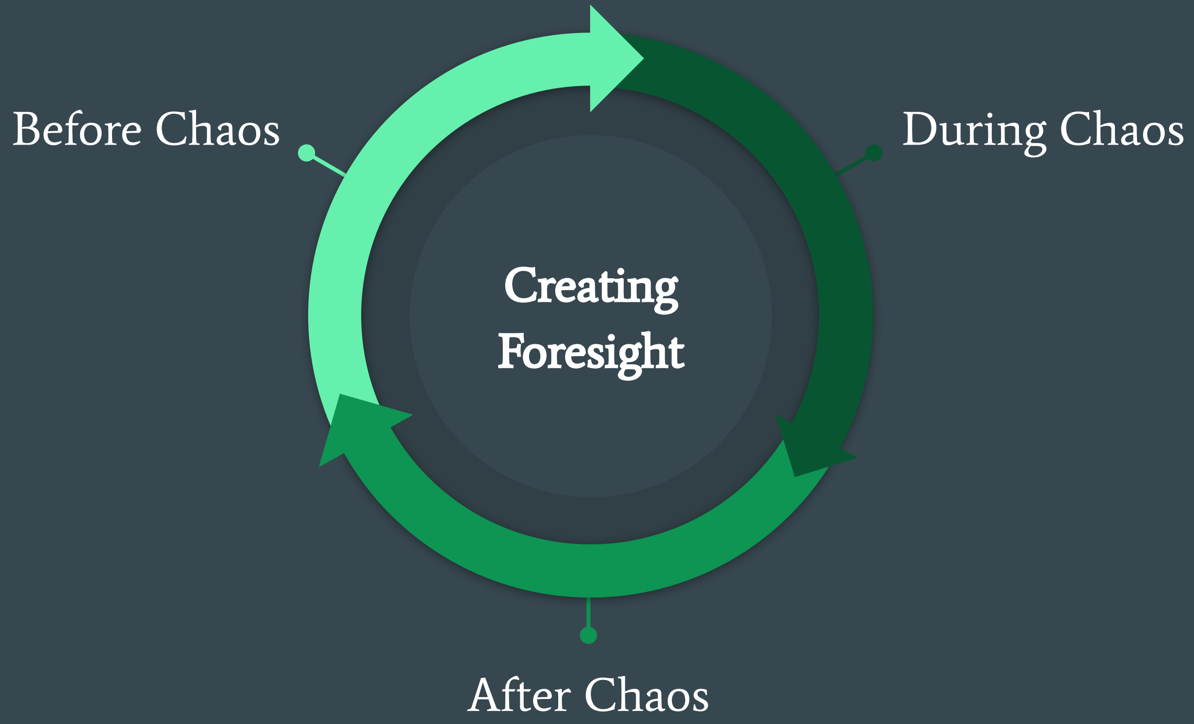
Chaos Engineering

Building Confidence in System Behavior through Experiments



AWS re:Invent 2017 - Nora Jones Describes Why We Need More Chaos - Chaos Engineering, That Is

Chaos Engineering is the **discipline of experimenting** *on a system in order to* **build confidence** *in the system's capability to* **withstand turbulent conditions** *in production.*



Creating Foresight

Before Chaos

During Chaos

After Chaos

@nora_js

Setting Common Ground

The goal of Chaos Engineering isn’t to use or build tooling to stop issues or find vulnerabilities for us.

The goal of Chaos Engineering isn’t to use or build tooling to stop issues or find vulnerabilities for us.

The goal is to build a culture of resilience to the unexpected. If tools can **help** with this, excellent…but don’t forget the main goal of chaos **as** you’re building these tools.

“Resilience is not about reducing errors. It’s about enhancing the positive capabilities of people and organizations that allow them to adapt effectively and safely under pressure”

- Dekker, Woods, Cook

Chaos Engineering is one means to help us enhance.

The Before

Before the experiment

“All [mental] models are wrong, but some are useful”

- George Box

Story time

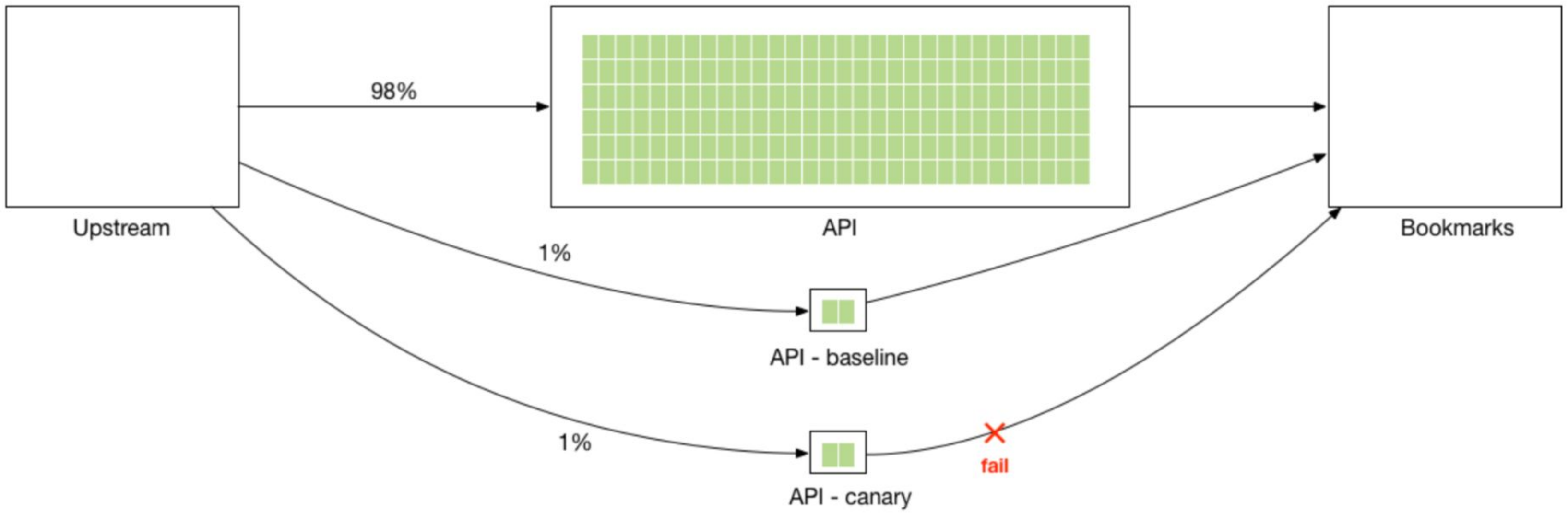


Fig. 6. ChAP experiment injecting failure into bookmarks from API

Source: *Automating Chaos Experiments in Production,* ICSE Report, 2019

@nora_js

Who was running these experiments?

Who was running these experiments? Mostly, just the chaos team.

So, what did we do?

We attempted to solve in a very common Software Engineer way...

We attempted to solve in a very common Software Engineer way...

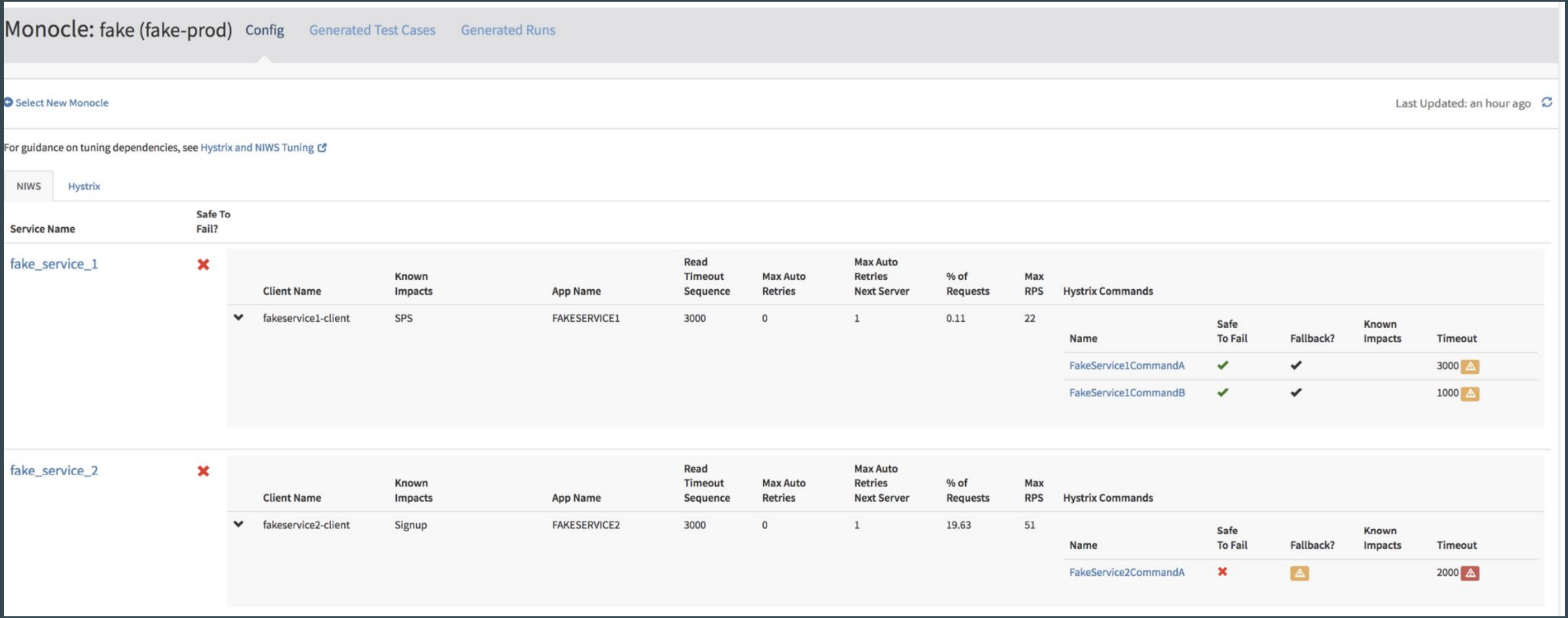
Any guesses on what we did?

We automated it!

A story on automating experiments

We had to collect information about services from a ton of different sources in order to do this:

- Timeouts
- Retries
- Fallbacks
- % of traffic it served



Monocle: fake (fake-prod) Config Generated Test Cases Generated Runs

Select New Monocle

Last Updated: an hour ago

For guidance on tuning dependencies, see Hystrix and NIWS Tuning

NIWS Hystrix

Service Name	Safe To Fail?
fake_service_1	✖
fake_service_2	✖

fake_service_1

Client Name	Known Impacts	App Name	Read Timeout Sequence	Max Auto Retries	Max Auto Retries Next Server	% of Requests	Max RPS
fakeservice1-client	SPS	FAKESERVICE1	3000	0	1	0.11	22

Hystrix Commands

Name	Safe To Fail	Fallback?	Known Impacts	Timeout
FakeService1CommandA	✓	✓		3000 ⚠
FakeService1CommandB	✓	✓		1000 ⚠

fake_service_2

Client Name	Known Impacts	App Name	Read Timeout Sequence	Max Auto Retries	Max Auto Retries Next Server	% of Requests	Max RPS
fakeservice2-client	Signup	FAKESERVICE2	3000	0	1	19.63	51

Hystrix Commands

Name	Safe To Fail	Fallback?	Known Impacts	Timeout
FakeService2CommandA	✖	⚠		2000 ⚠

Source: *Automating Chaos Experiments in Production*, ICSE Report, 2019

The most insightful part of doing the Chaos Engineering experiment is not the experiment itself, but **the process of designing** it.

How do we design a meaningful experiment?

Pick a facilitator.

What do we want to experiment on?

Involve all people participating in the experiment preparation.

The Apollo 1 Launch Rehearsal Test

@nora_js

Apollo 1 Launch Rehearsal Test

Not all of the necessary parties were involved in the design of the experiment (like a rescue team or medical assistance team).

Novices vs. Experts

“The new employee could see the times when the system **broke down**, but not the times when it worked.”

- Seeing the Invisible, Klein and Hoffman

@nora_js

Start looking at your incident data and make some observations.

If your incident data resembles the following:

- MTTR
- MTTD
- Date/Time
- Pager ID
- Severity
- Root Cause

…..

You're going to want to spend some time gathering **more** meaningful data that helps you understand where gaps are.

You can derive more meaningful experiments with **this** kind of data:

- Which systems **haven’t** failed in awhile?
- Which systems had failures **that took us by surprise?**
- Which incidents **involved “un-owned” systems** or ones that needed an “expert” to step in and resolve?
- Which incidents involved people that **hadn’t worked together much?**
- What did recent **“near misses”** look like?
- Which incidents involved **difficulties in figuring out what was even going on?**

How teams decide what to experiment on, is just as revealing as the experiment itself.

Discuss Scope

Once you've chosen an experiment:

- **How** do you define a "normal" or "good" operation?
- **How** are we establishing the *scope (where we are injecting the failure)* of the experiment

Discuss Scope

- **What** do we expect to happen with this failure? *(be explicit)*
  - What do we expect from individual components?
  - What do we expect from the system as a whole?

Discuss Scope

- **How** do you know if the system is in a bad state?
    - Which metrics are most important for us to measure during the experiment?
    - “Observability is **feedback that provides insight in to a process** and refers to the work needed to extract meaning from available data.” - Woods & Hollnagel, Joint Cognitive Systems
    - With this in mind, how are we **observing** the system?

Discuss Scope

- **How** are we limiting the blast radius?

Discuss Scope

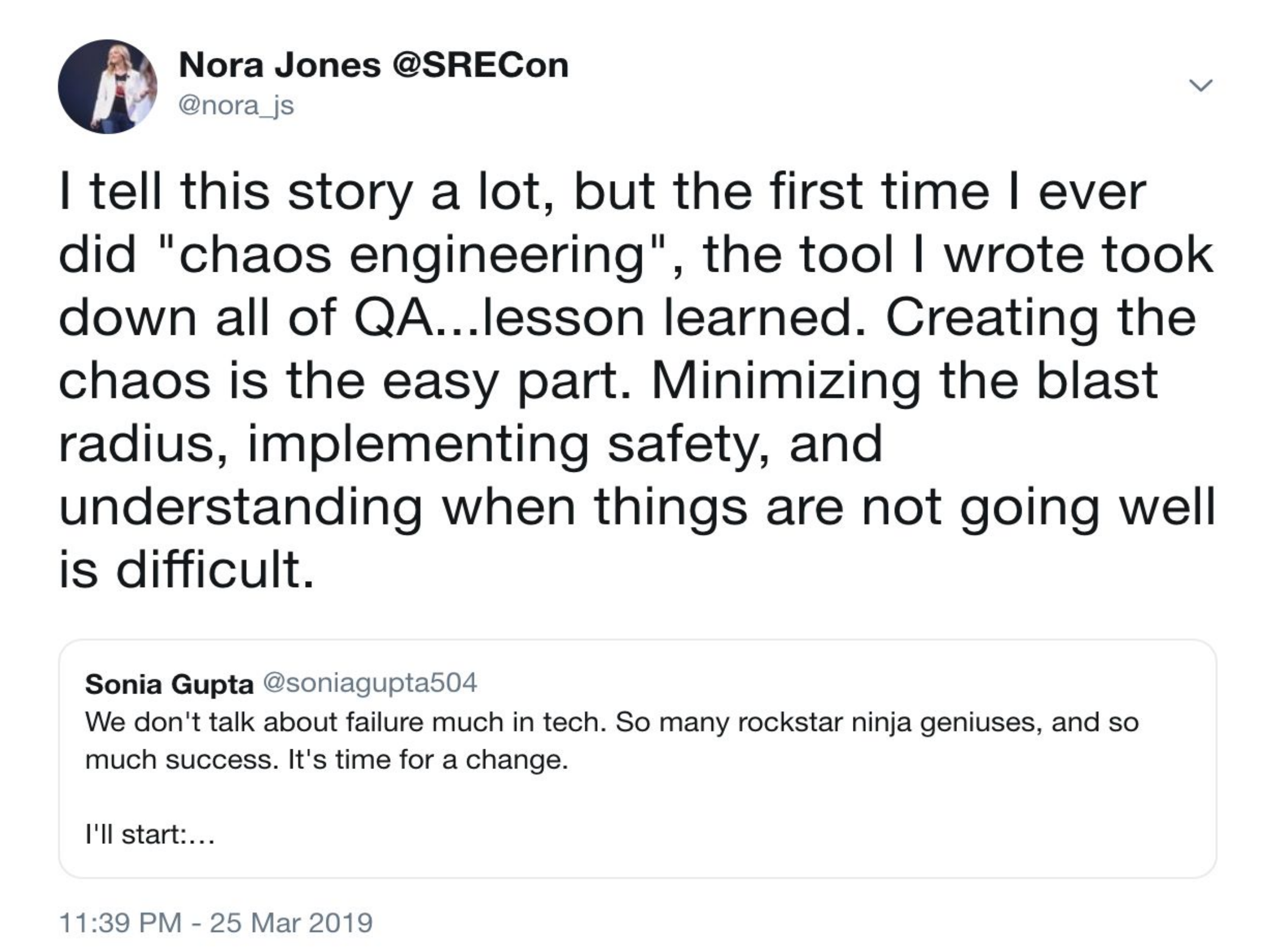
- **What** is the perceived business value of the experiment (to the service team)?
    - What is the perceived value to the rest of the organization? *(Note: these might be different, and that's ok)*

@nora_js

Hypothesize

The During

@nora_js



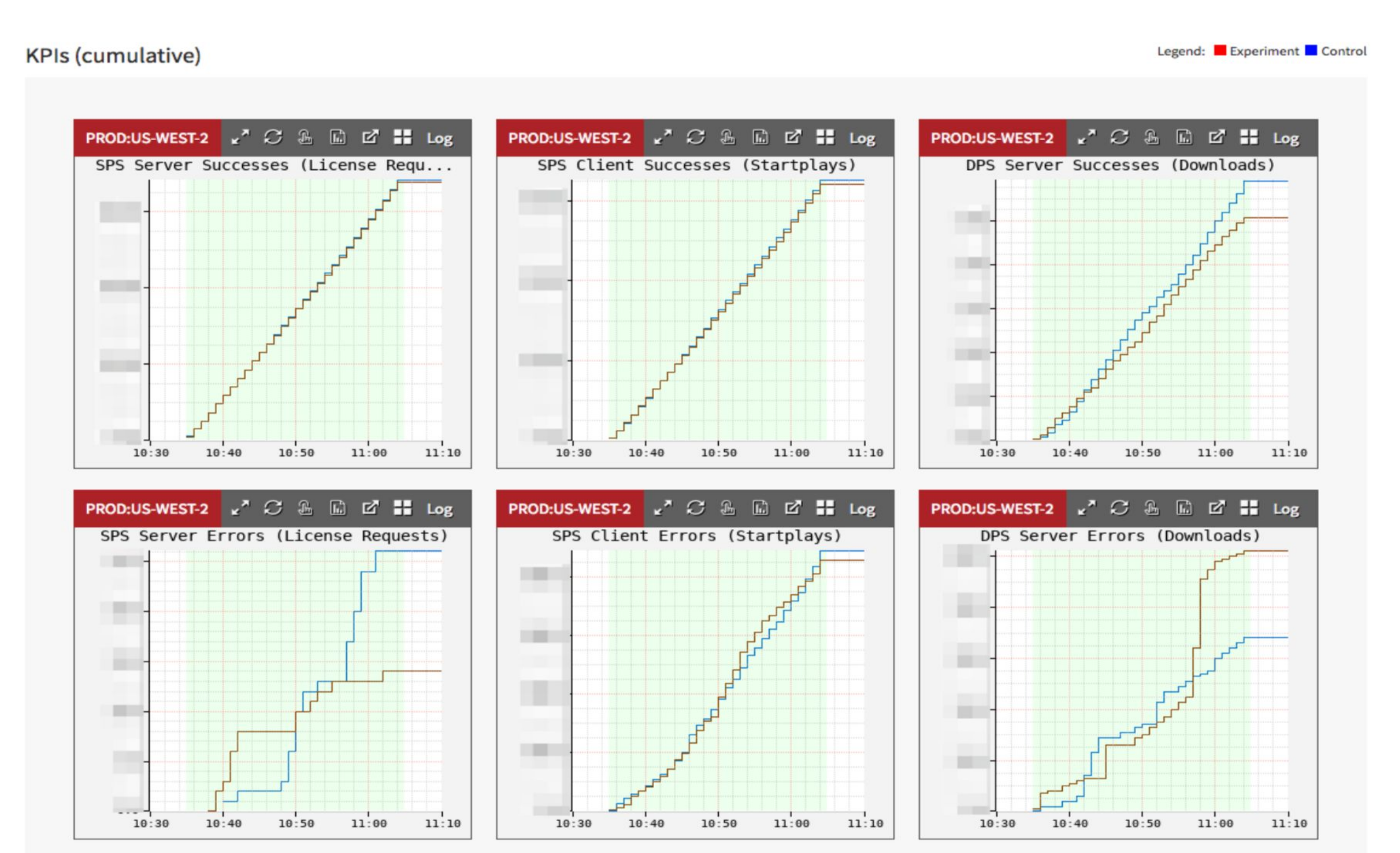
Nora Jones @SRECon
@nora_js

I tell this story a lot, but the first time I ever did "chaos engineering", the tool I wrote took down all of QA...lesson learned. Creating the chaos is the easy part. Minimizing the blast radius, implementing safety, and understanding when things are not going well is difficult.

> **Sonia Gupta** @soniagupta504
> We don't talk about failure much in tech. So many rockstar ninja geniuses, and so much success. It's time for a change.
>
> I'll start:...

11:39 PM - 25 Mar 2019

@nora_js


KPIs (cumulative)
Legend: 🟥 Experiment 🟦 Control
PROD:US-WEST-2
Log
SPS Server Successes (License Requ...
SPS Client Successes (Startplays)
DPS Server Successes (Downloads)
SPS Server Errors (License Requests)
SPS Client Errors (Startplays)
DPS Server Errors (Downloads)
10:30
10:40
10:50
11:00
11:10

Source: *Automating Chaos Experiments in Production,* ICSE Report, 2019

Roles

- **Designer** (the person leading the discussion)
- **Facilitator** (the person keeping the group on track)
- **Commander** (the person executing the commands)
- **Scribe** (takes notes in a communication tool (such as Slack or GDocs) on what is occurring in the room)
- **Observer** (looks at and shares relevant graphs with the rest of the group)
- **Correspondent** (keeps an eye on #alerts-type-channel and makes sure the current on-call is aware of the experiment occurring and what the expected impact is)

Steady State

“Start by defining steady state as some measurable output of a system that indicates normal behavior.”

- principlesofchaos.org

Don’t be lenient about the definition of the ’steady state’.

Apollo 1 Launch Rehearsal on: Steady State

“**The morning of the test**, the crew suited up and **detected a foul odor in the breathing oxygen**, which took about an hour to fix. Then the **communications system acted up**. Shouting through the noise, Grissom vented: "How are we going to get to the moon if we can't talk between two or three buildings?"

- Elizabeth Howell @HowellSpace

The After

“Resilience is the story of the outage that never happened”

- John Allspaw

If “resilience is the story of the outage that never happened”…

If “resilience is the story of the outage that never happened”…

what story are you telling and **who** is there to hear it?

Asking better questions will lead to better stories.

Forget templates, ask questions instead

- **What** did you experiment on?
- **Why** did you experiment on it?
- **What** were the reactions?
- **What** mental models got recalibrated?
- **What** was surprising?
- **What** was new?

Forget templates, ask questions instead

- **Who** was present?
- **How** did we assess ourselves?
- **What** are the relationships between the people that are present?
- **When** was the last time they worked together in an incident?
- **Did** the system gracefully degrade or fully collapse at any point?
- **What** were the necessary adaptations?

Forget templates, ask questions instead

- **How** long did it take to find the right graphs?

@nora_js

Looking at what went right in a chaos experiment, helps us understand what went wrong.

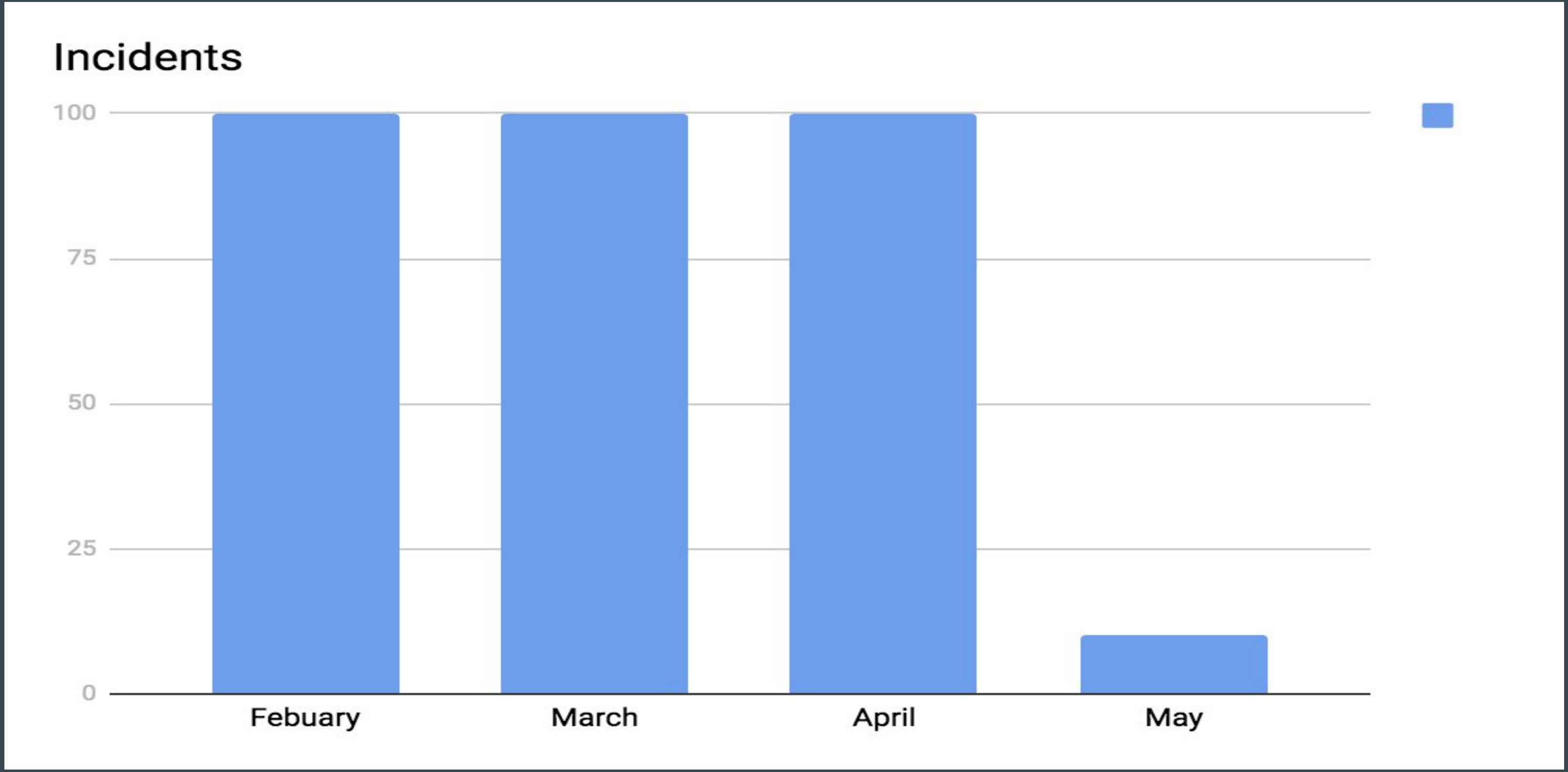
“The pursuit of success”… “what did we learn we are really good at” - Sidney Dekker

We can leverage Chaos Engineering to measure sources of resilience, and create feedback loops that enhance the organization’s ability to:

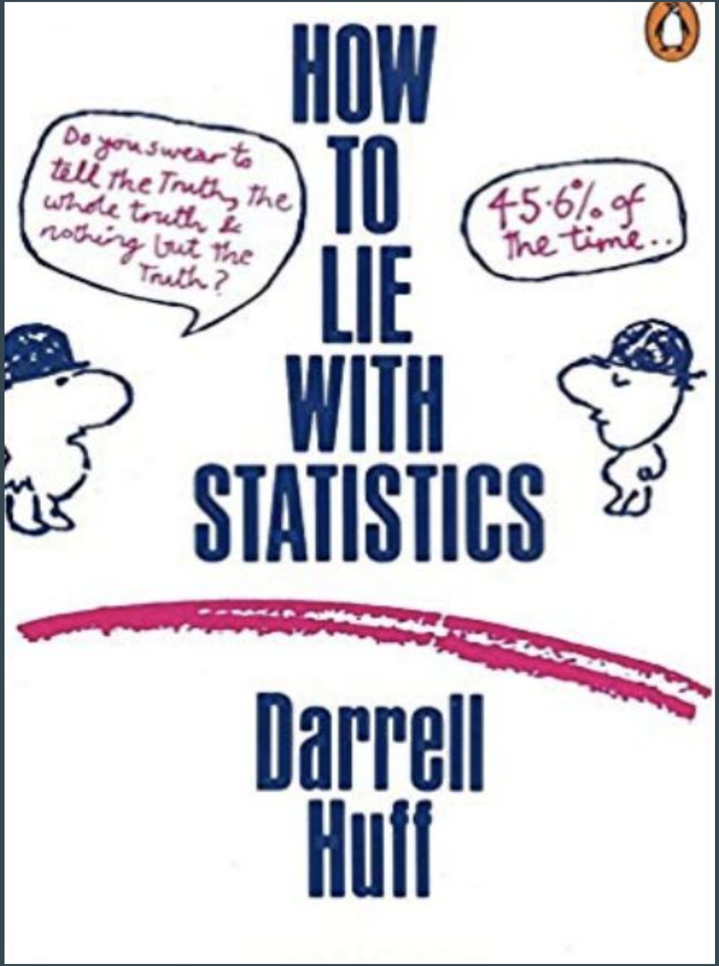
1. Monitor and revise mental models
2. Target safety investments
3. Build expertise

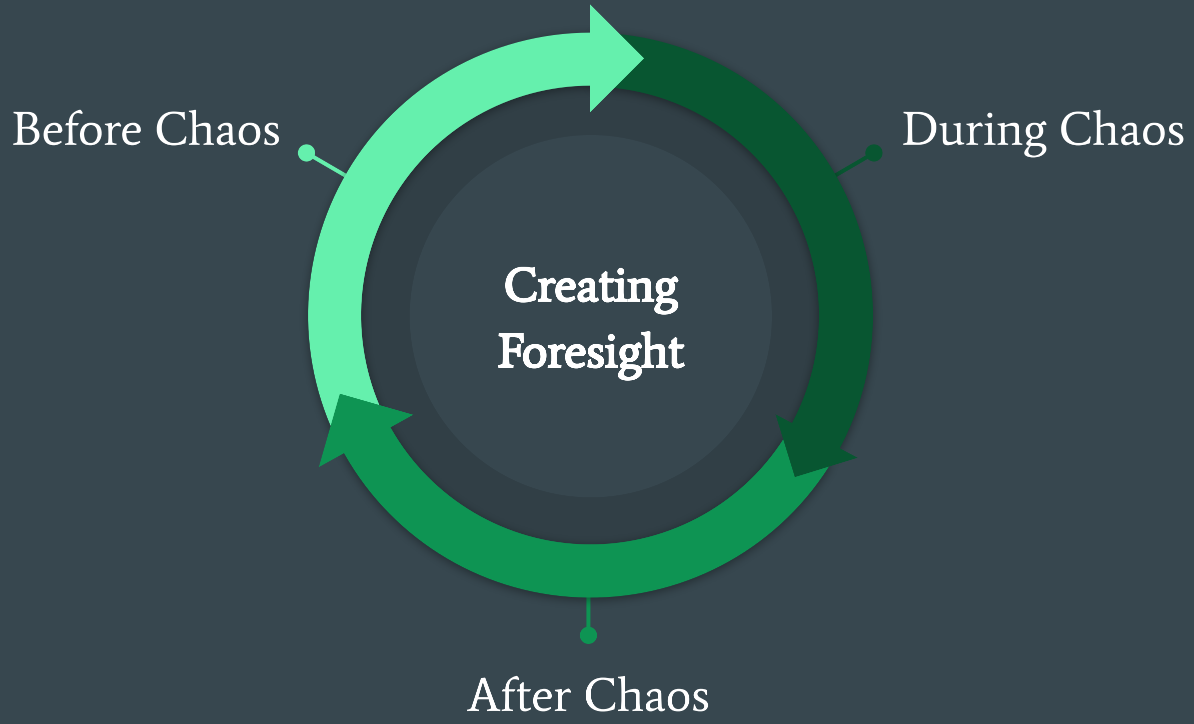
Chaos Engineering reports can be used as a cultural artifact. You can give these to new employees as a way to understand how people work together under pressure and when the system breaks down.

This is not learning:



@nora_js


HOW TO LIE WITH STATISTICS
Do you swear to tell the Truth, the whole truth & nothing but the Truth?
45·6% of the time..
Darrell Huff



@nora_js

Key Takeaways: Creating Foresight

- You can’t shortcut some of the work involved here
- Safety shouldn’t just fall on ops or reliability-focused teams
- Don’t give new employees vulnerabilities to go through and fix
- Use write-ups as learning opportunities
- Connect all phases of chaos and include incident data as well