Learning Through Failure

Rich Burroughs
Community Manager
Gremlin, Inc.
@richburroughs

Dream of the '90s | Portlandia | IFC - YouTube

https://www.youtube.com/watch?v=U4hShMEk1Ew

Complexity is constantly increasing

WebMD
https://web.archive.org/web/20000229190054/http://www.webmd.com/
INTERNET ARCHIVE WaybackMachine
http://www.webmd.com/
Go
70,595 captures
28 Dec 1996 – 6 Apr 2019
NOV 1999
FEB 29 2000
APR 2001
About this capture

press release
America Speaks Out
announcement →

WebMD℠
Health has a *Homepage.*

From our partner Medtronic
SUDDEN CARDIAC ARREST
Click here to beat the odds →

from the WebMD National News Center

Prospectors Mine Nature for New Drugs

In the Old West, the first glimmer of gold in a stream lured prospectors from across the continent with the promise of instant riches. Now, new prospectors called bioprospectors are looking for the source of the next major drugs to treat cancer, AIDS, and a host of other diseases.

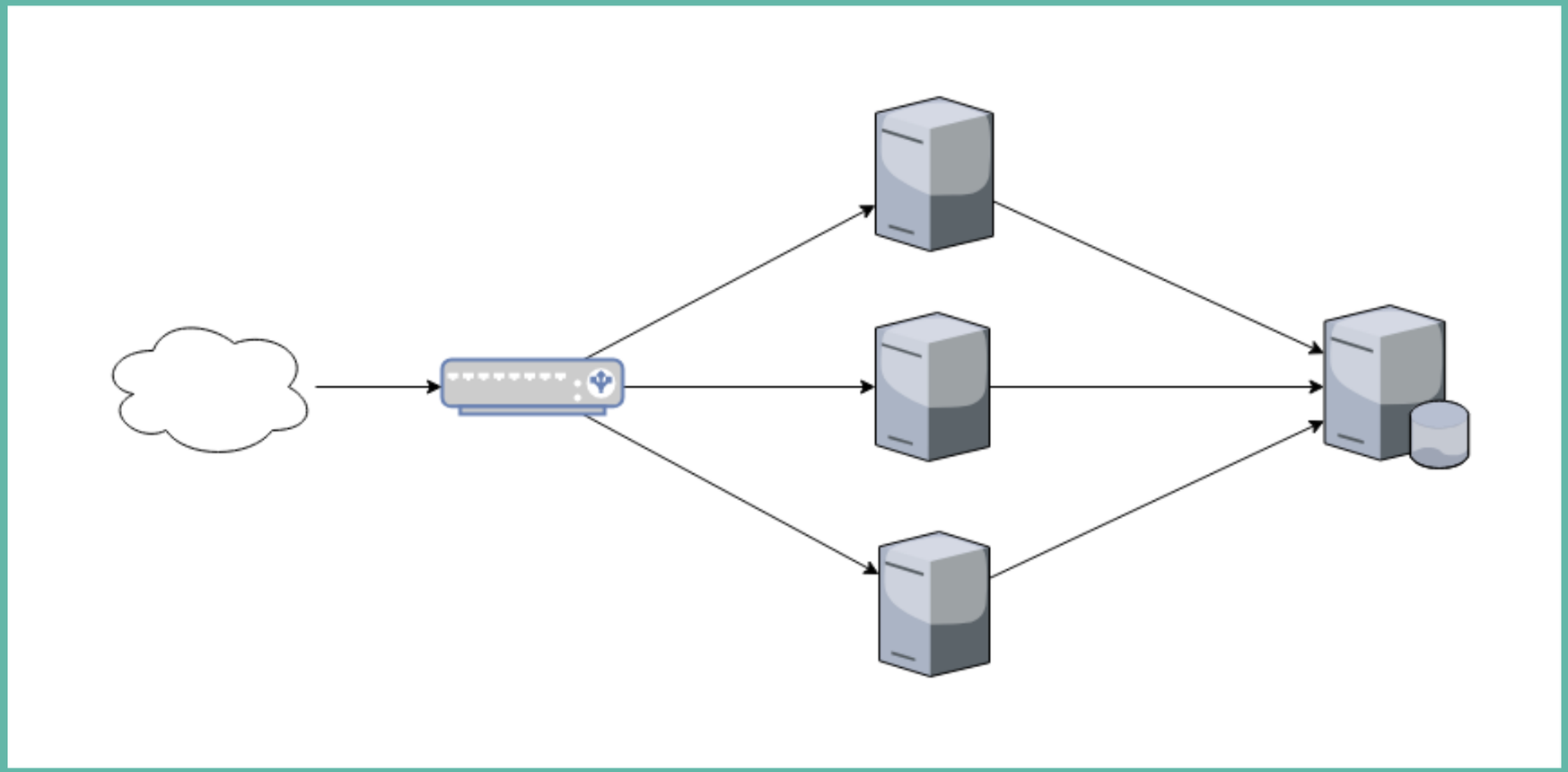
Go to Consumer Home and News

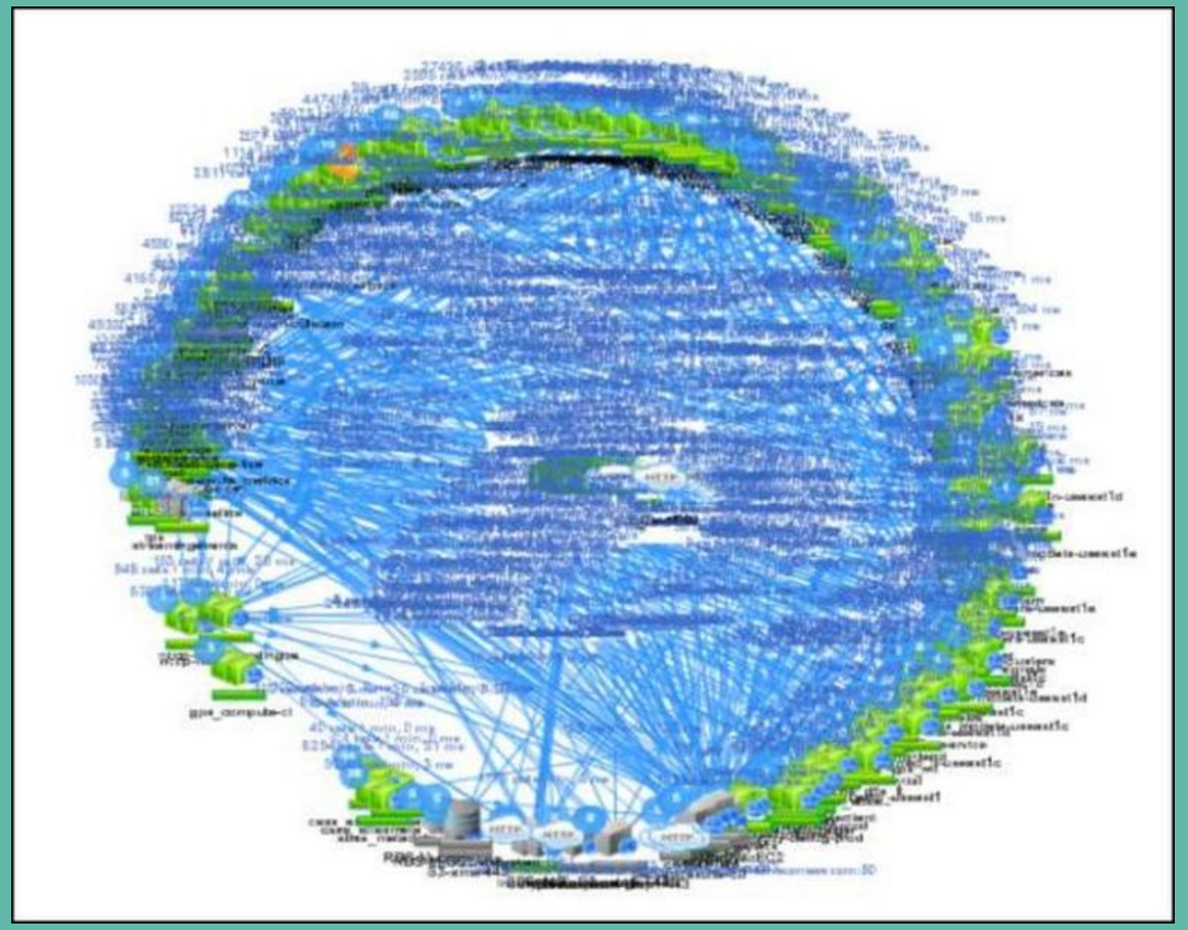
enter CONSUMER site
PHYSICIAN — Enter Members — Enter Non-Members
HEALTH TEACHER
WebRN
OFFICE MANAGER

About Healtheon | WebMD
BenefitCentral

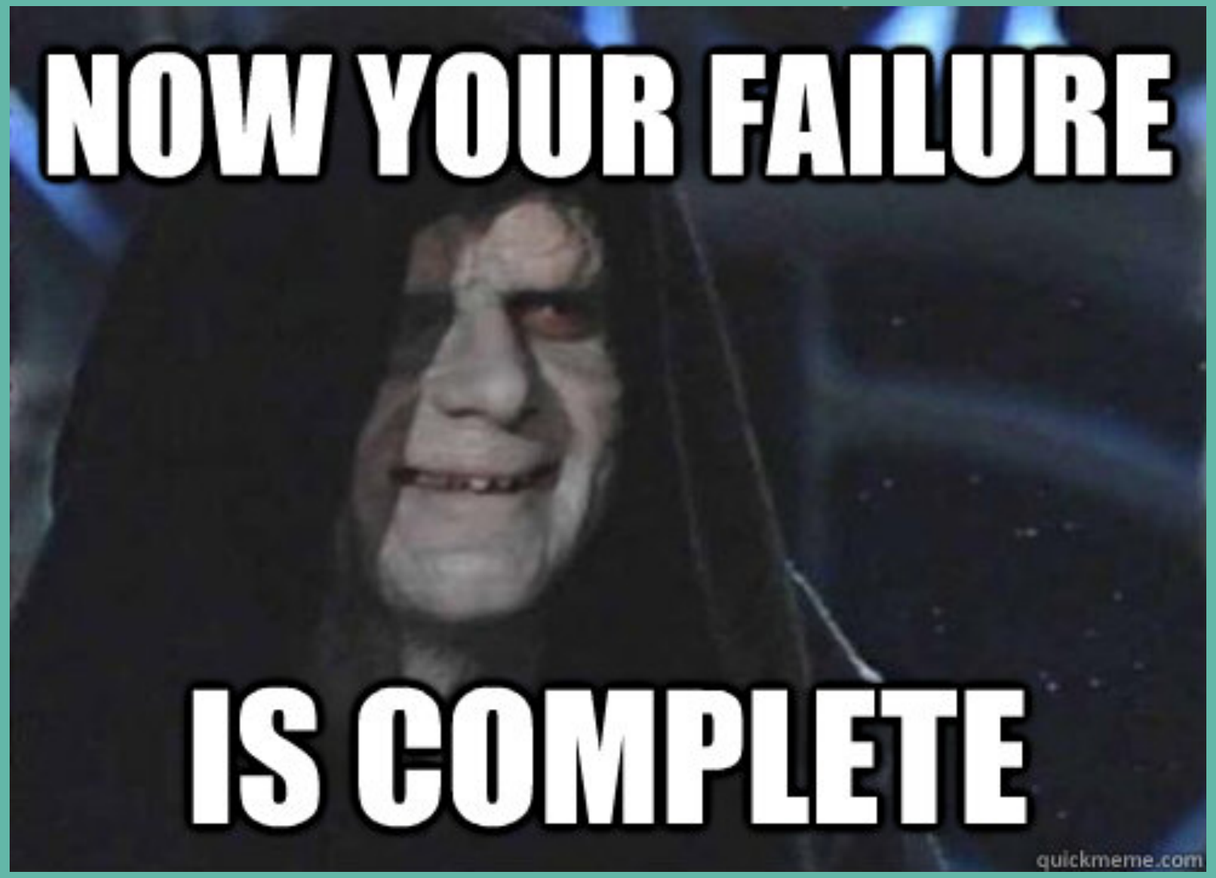
NEW! Take a **sneak peek** at our **Physician site** and see how we're using the **Internet** to **revolutionize** the world of **medicine**

Copyright © 2000 Healtheon/WebMD Corporation - All rights reserved.





What's changed?


NOW YOUR FAILURE
IS COMPLETE
quickmeme.com

How Complex Systems Fail

(Being a Short Treatise on the Nature of Failure; How Failure is Evaluated; How Failure is Attributed to Proximate Cause; and the Resulting New Understanding of Patient Safety)

Richard I. Cook, MD
Cognitive technologies Laboratory
University of Chicago

1) **Complex systems are intrinsically hazardous systems.**
All of the interesting systems (e.g. transportation, healthcare, power generation) are inherently and unavoidably hazardous by the own nature. The frequency of hazard exposure can sometimes be changed but the processes involved in the system are themselves intrinsically and irreducibly hazardous. It is the presence of these hazards that drives the creation of defenses against hazard that characterize these systems.

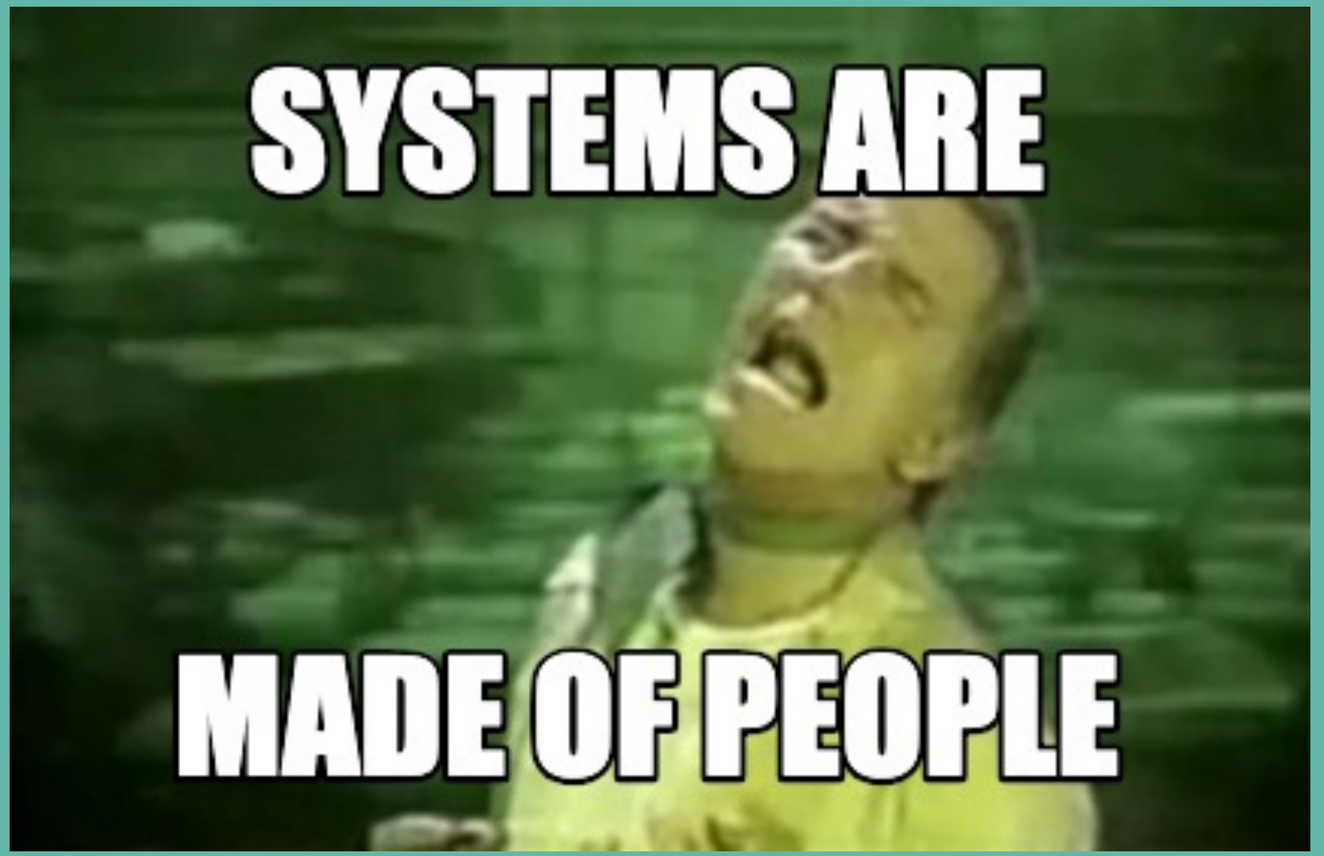
2) **Complex systems are heavily and successfully defended against failure.**
The high consequences of failure lead over time to the construction of multiple layers of defense against failure. These defenses include obvious technical components (e.g. backup systems, 'safety' features of equipment) and human components (e.g. training, knowledge) but also a variety of organizational, institutional, and regulatory defenses (e.g. policies and procedures, certification, work rules, team training). The effect of these measures is to provide a series of shields that normally divert operations away from accidents.

3) **Catastrophe requires multiple failures – single point failures are not enough..**
The array of defenses works. System operations are generally successful. Overt catastrophic failure occurs when small, apparently innocuous failures join to create opportunity for a systemic accident. Each of these small failures is necessary to cause catastrophe but only the combination is sufficient to permit failure. Put another way, there are many more failure opportunities than overt system accidents. Most initial failure trajectories are blocked by designed system safety components. Trajectories that reach the operational level are mostly blocked, usually by practitioners.

"Catastrophe is always just around the corner"

"Change introduces new forms of failure"

"All practitioner actions are gambles"


SYSTEMS ARE
MADE OF PEOPLE

John Allspaw
@allspaw

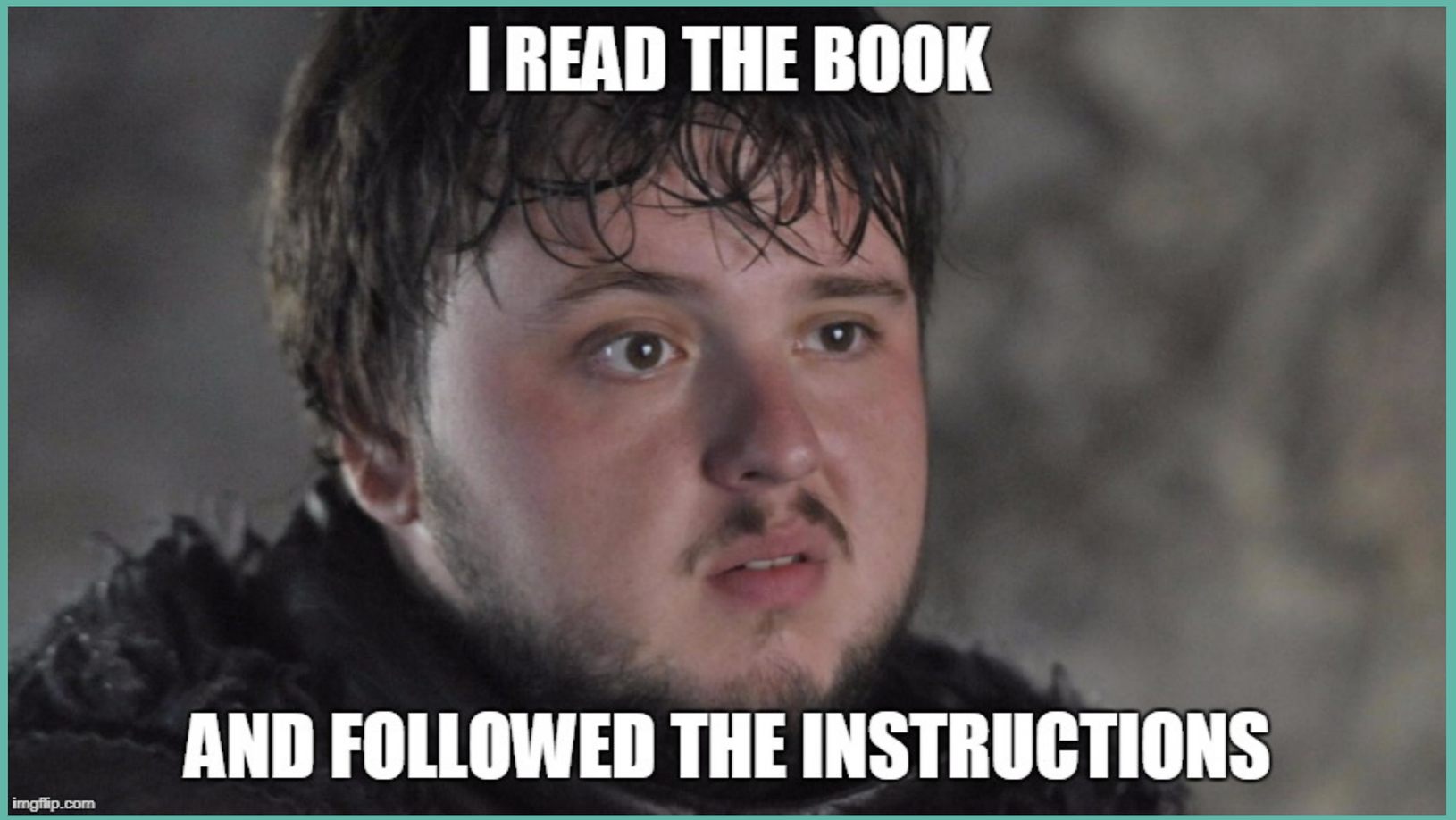
Following

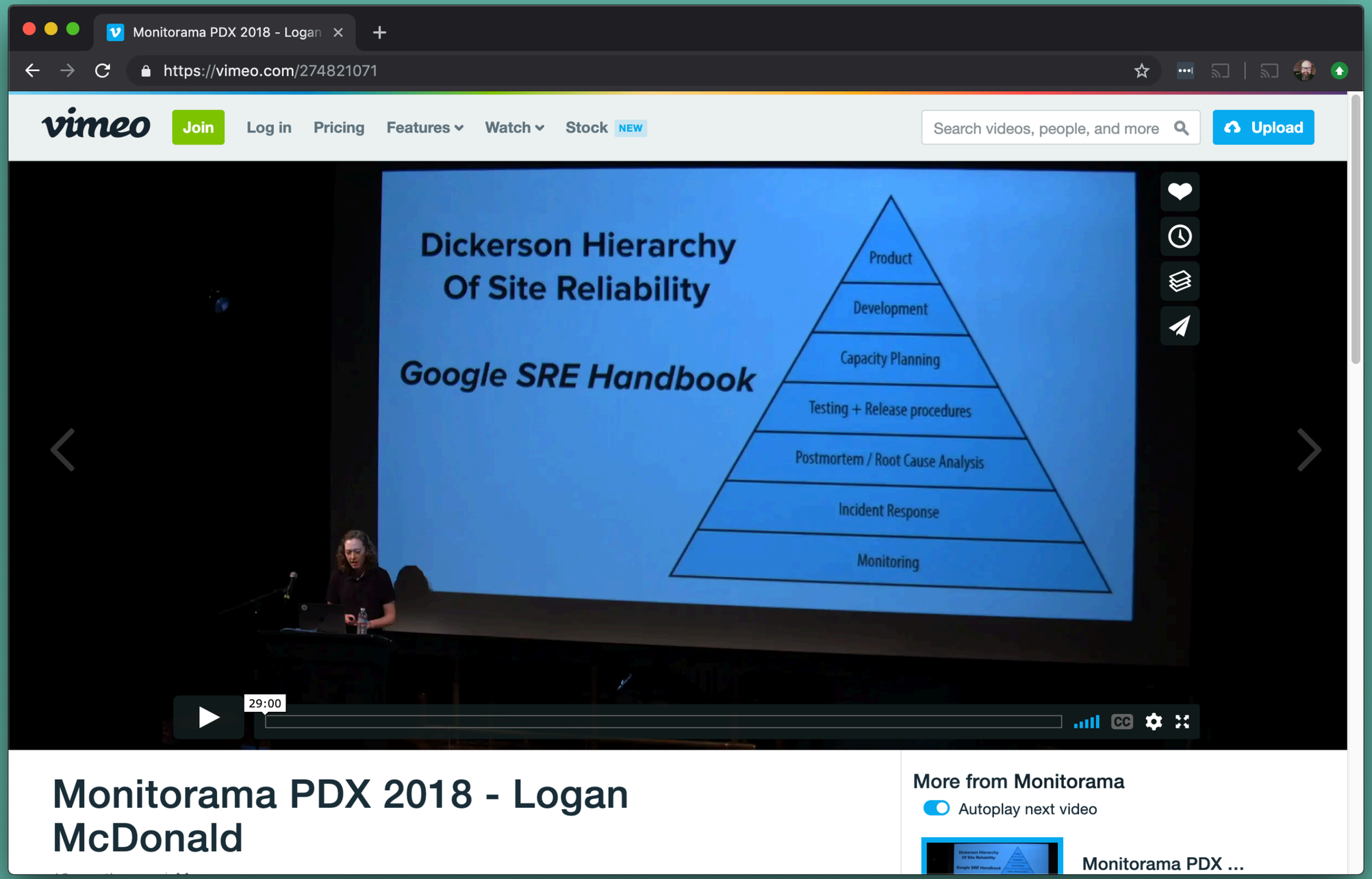
Replying to @copyconstruct @ajdomie @mipsytipsy

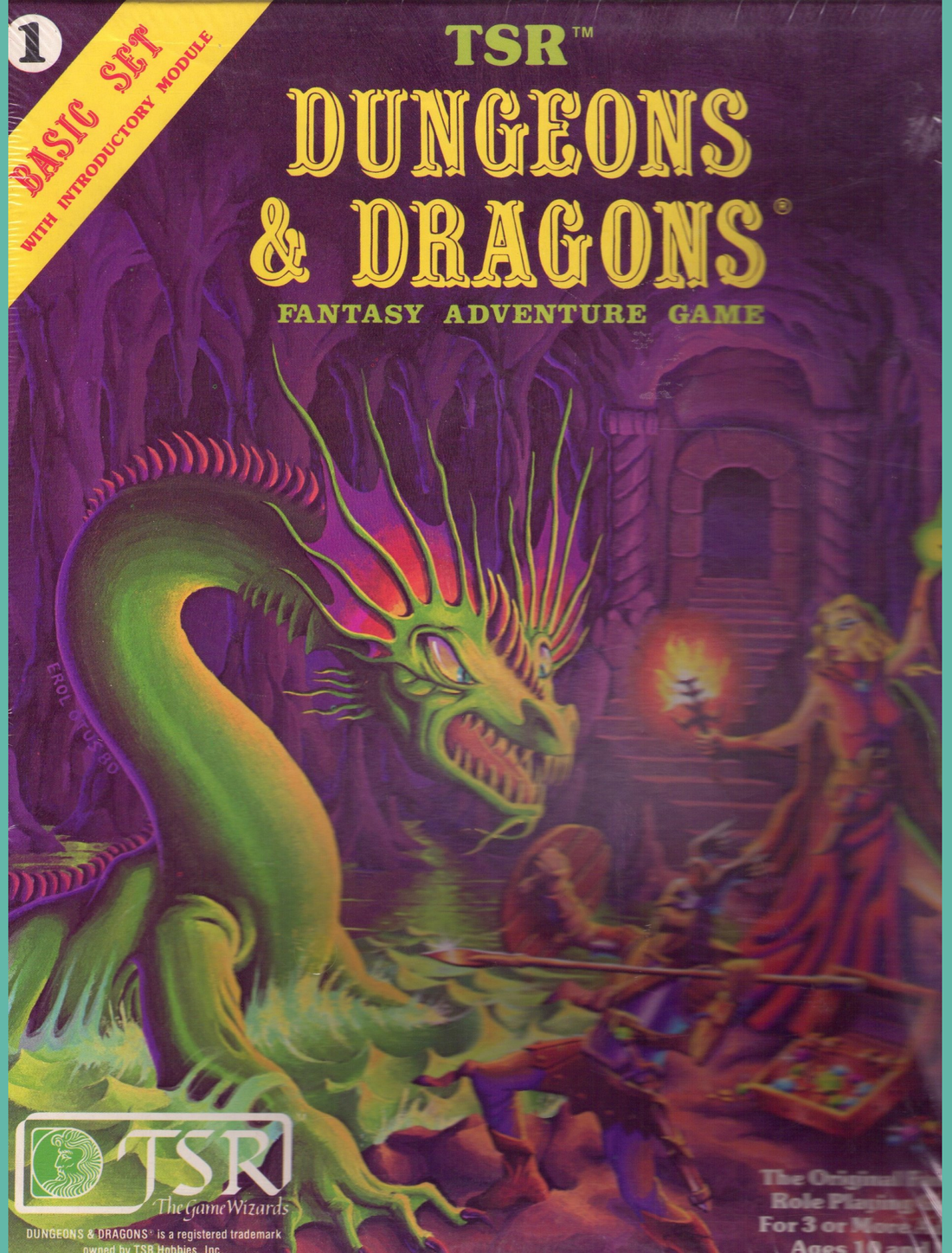
This plot is a fine companion to “Woods' Theorem: As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases rapidly.”

9:16 AM - 14 Feb 2018

What are some ways we can learn more about systems?


I READ THE BOOK
AND FOLLOWED THE INSTRUCTIONS
imgflip.com


Monitorama PDX 2018 - Logan
https://vimeo.com/274821071
vimeo
Join
Log in
Pricing
Features
Watch
Stock
NEW
Search videos, people, and more
Upload
Dickerson Hierarchy
Of Site Reliability
Google SRE Handbook
Product
Development
Capacity Planning
Testing + Release procedures
Postmortem / Root Cause Analysis
Incident Response
Monitoring
29:00
CC
Monitorama PDX 2018 - Logan McDonald
More from Monitorama
Autoplay next video
Monitorama PDX ...


1
BASIC SET
WITH INTRODUCTORY MODULE
TSR™
DUNGEONS
& DRAGONS®
FANTASY ADVENTURE GAME
TSR
The Game Wizards
DUNGEONS & DRAGONS® is a registered trademark
owned by TSR Hobbies, Inc.

Chaos Engineering

"Thoughtful, planned experiments designed to reveal weakness in our systems."

Home - Chaos Monkey

https://netflix.github.io/chaosmonkey/

Chaos Monkey

Search docs

- Home
- Configuration file format
- Configuring behavior via Spinnaker
- Decryptor
- Error counter
- How to deploy
- Outage checker
- Running locally
- Running tests
- Termination behavior
- Tracker
- Vendoring dependencies

GitHub Next »

Docs » Home

Edit on GitHub

Chaos Monkey is responsible for randomly terminating instances in production to ensure that engineers implement their services to be resilient to instance failures.

See how to deploy for instructions on how to get up and running with Chaos Monkey.

Once you're up and running, see configuring behavior via Spinnaker for how users can customize the behavior of Chaos Monkey for their apps.

Next

A Netflix Original Production

Netflix OSS | Tech Blog | Twitter @NetflixOSS | Jobs

Built with MkDocs using a theme provided by Read the Docs.

Prerequisites

— Observability
— Blameless Culture

Scientific Method

- Ask a question
- Research
- Form a hypothesis
- Experiment to test the hypothesis
- Analyze data and draw a conclusion
- Share the results

Types of attacks

- Shutdown
- CPU
- Memory
- I/O
- Network Latency
- Packet Loss
- DNS
- Blackhole


38
AND BOOM GOES THE DYNAMITE
memegenerator.net

The goal is to experiment in Production

CHAOS EXPERIMENT FORM

APPLICATION NAME

MONITORING TOOL

EXPERIMENT NAME

FAILURE TO INJECT

IMPACT OF FAILURE

SCOPE OF FAILURE

DURATION OF FAILURE

TOOL TO USE

HYPOTHESIS

RESULTS

ABORT CONDITION

ACTION ITEMS

JOIN US ON SLACK/CHAOSENGINEERING

Example experiment

- Application: Front End
- Attack: CPU
- Hypothesis: Adding CPU load will cause additional hosts to spin up in our Autoscaling Group
- Abort condition: Latency increases by 20%

Example experiment #2

- Application: Front End
- Attack: Blackhole
- Hypothesis: Blackholing the hostname for the Twilio API will cause the SMS transmissions to time out
- Abort condition: Error rate increases by 20%

Don't experiment on things you know are broken


BRACE YOURSELF
IT'S GAMEDAY
memes.com

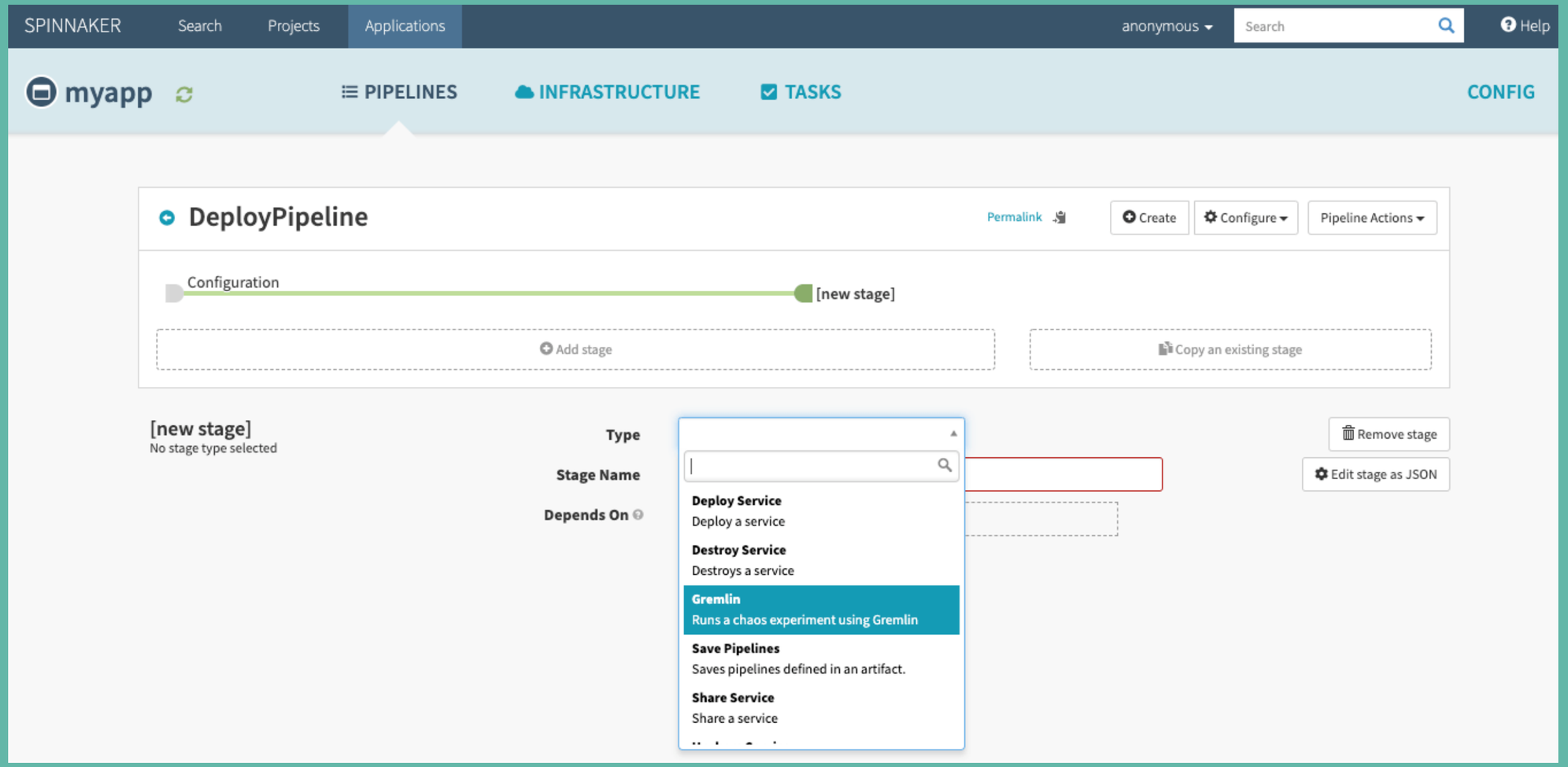
Questions

— Were we able to measure the results?
— Did the system respond the way we expected?
— Are there things we need to fix?

Run experiments to simulate an incident you've had

What comes after Game Days?

Continuous Chaos



Maturity model

- Running manual experiments
- Running experiments using Chaos Engineering tools
- Regularly scheduled Game Days
- Experimenting in Production
- Continuous Chaos

Next steps:

- Join our Chaos Engineering Slack: gremlin.com/slack
- Read tutorials: gremlin.com/community
- Chaos Conf: chaosconf.io
- Gremlin Free: go.gremlin.com/richchaos

Thank you!

Twitter: @richburroughs
Email: richb@gremlin.com

Slides: https://github.com/richburroughs/learningthroughfailure