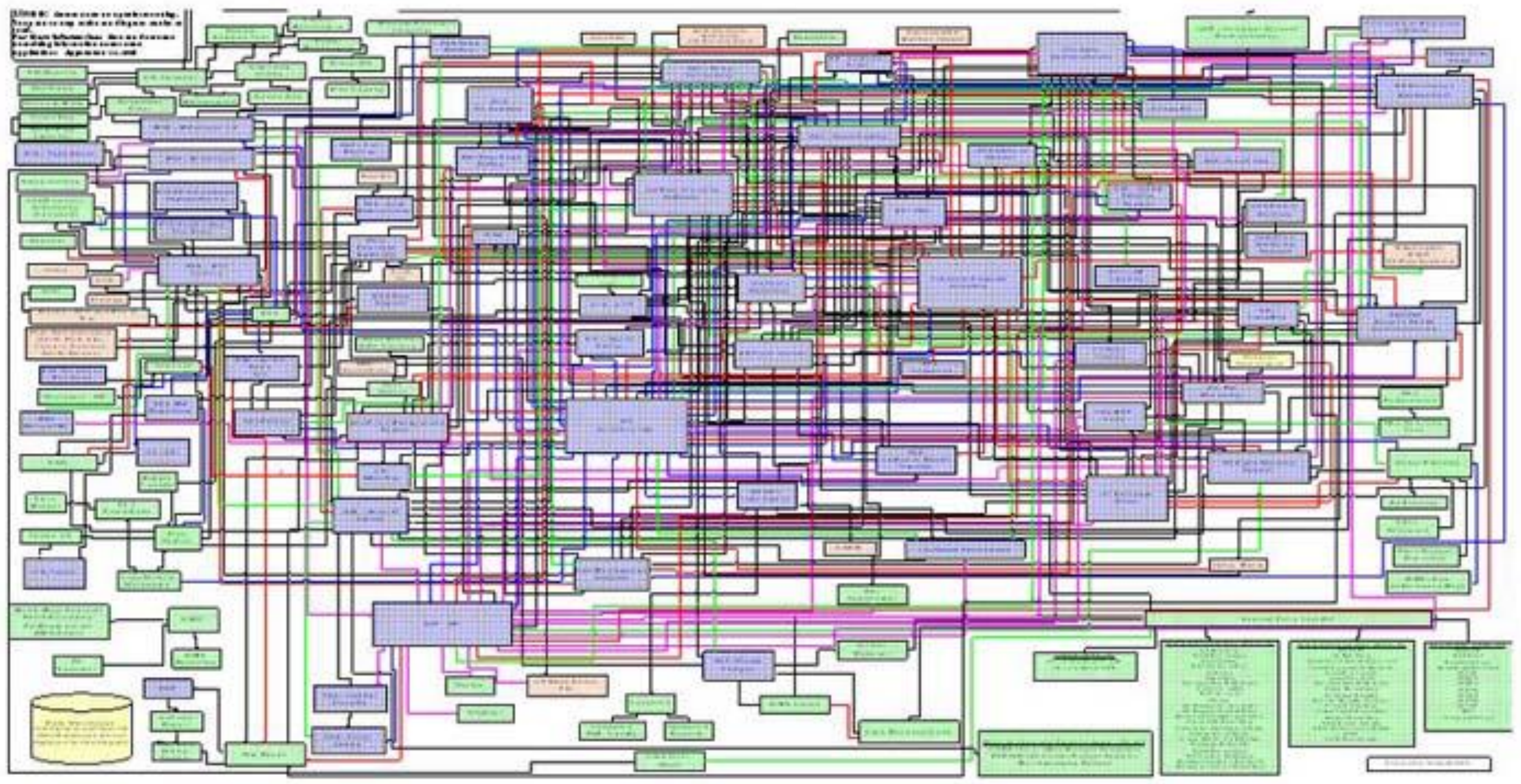
THE CASE FOR CHAOS TESTING

Peter Lamar
VP Cloud & Developer Relations
Peter.Lamar@Softwareag.com
peterrlamar@gmail.com
@ptlamar

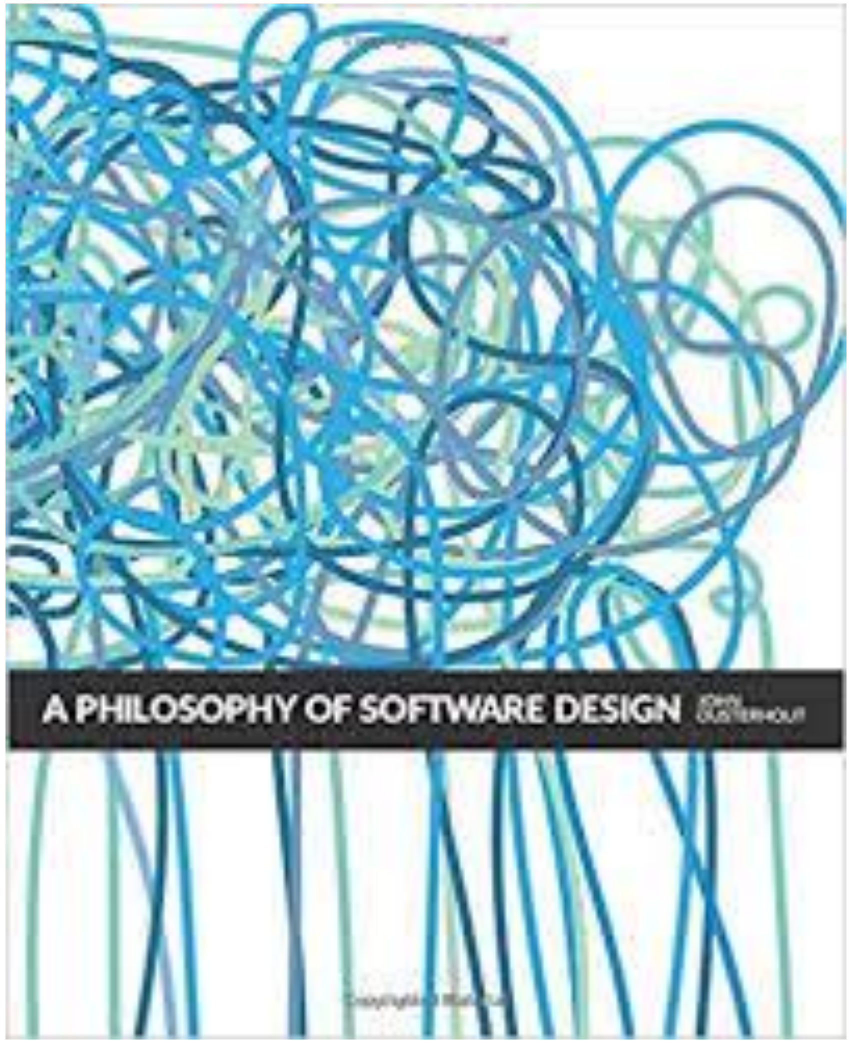
COMPLEXITY



COMPLEXITY

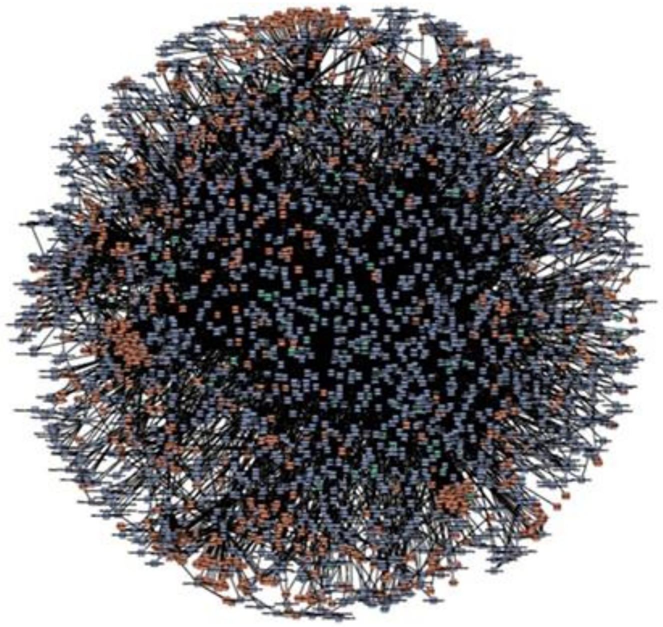
Its easy to build software that becomes complex

- Fight complexity one line at a time



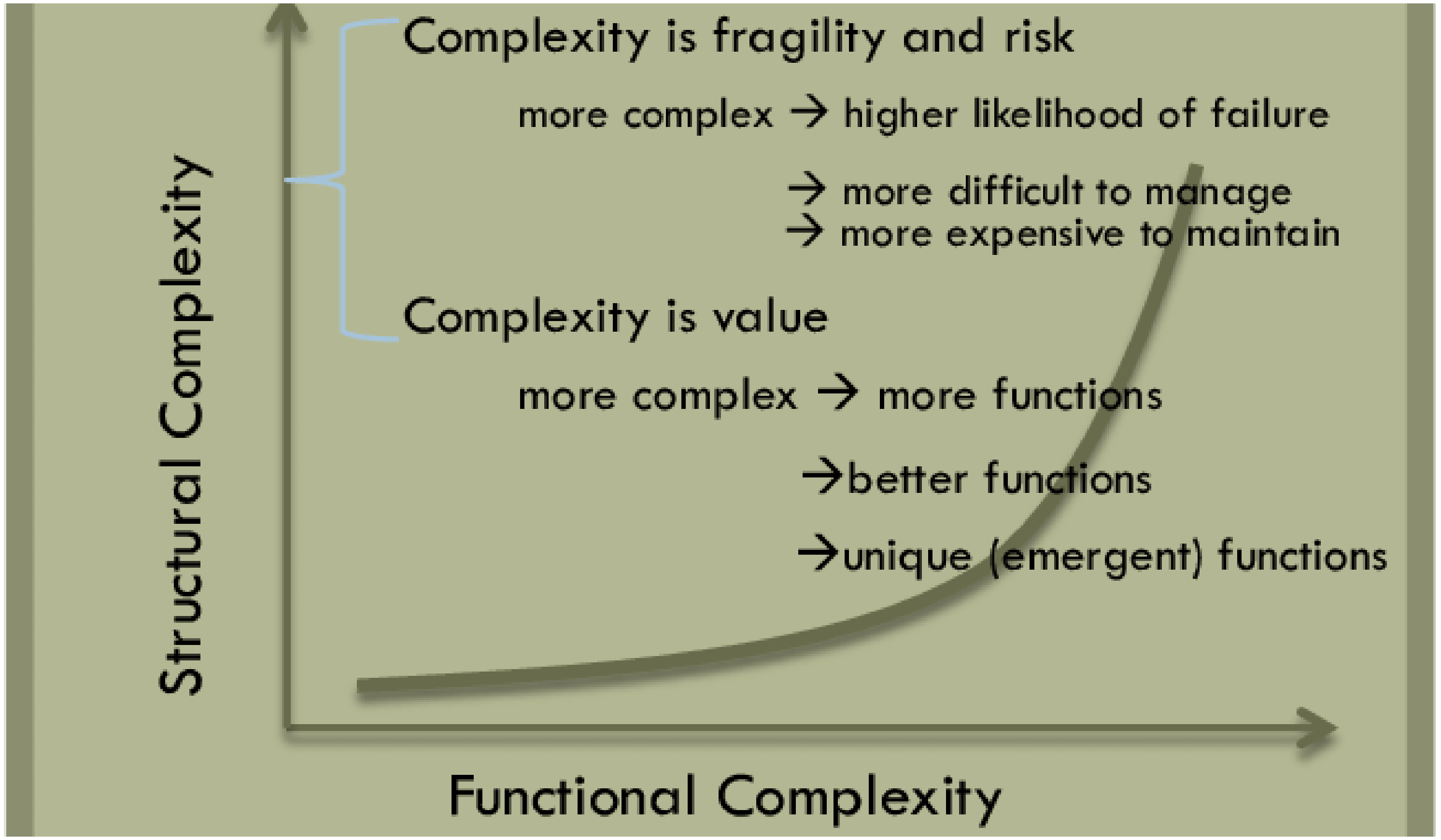
COMPLEXITY

However, even the best systems grow in complexity over time



amazon.com

NETFLIX


Complexity is fragility and risk
more complex → higher likelihood of failure
→ more difficult to manage
→ more expensive to maintain
Complexity is value
more complex → more functions
→better functions
→unique (emergent) functions
Structural Complexity
Functional Complexity

COMPLEXITY

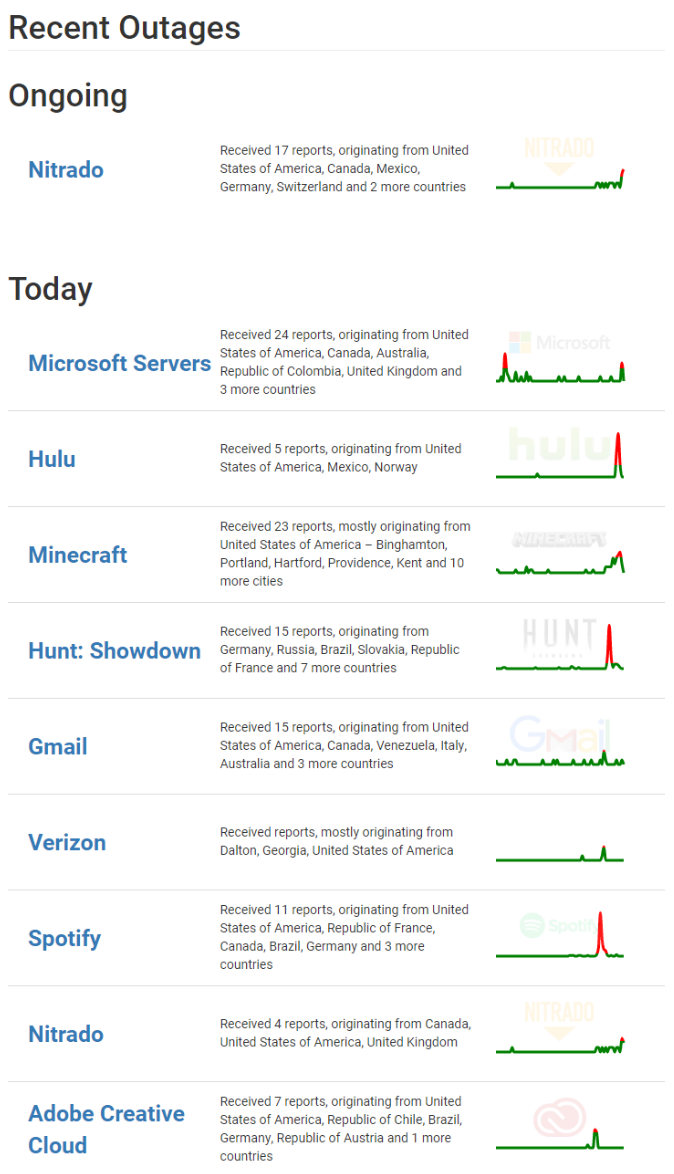
Systems will grow in complexity, which will raise the likelihood of failure unless steps are taken to mitigate and manage complexity

COMPLEXITY

Q. Do more professional, larger and better funded teams have less failure? i.e. Imposter syndrome?

A. Lets find out at https://outage.report

(*Screenshot 8/22/19)



Recent Outages

Ongoing

Nitrado	Received 17 reports, originating from United States of America, Canada, Mexico, Germany, Switzerland and 2 more countries

Today

Microsoft Servers	Received 24 reports, originating from United States of America, Canada, Australia, Republic of Colombia, United Kingdom and 3 more countries
Hulu	Received 5 reports, originating from United States of America, Mexico, Norway
Minecraft	Received 23 reports, mostly originating from United States of America – Binghamton, Portland, Hartford, Providence, Kent and 10 more cities
Hunt: Showdown	Received 15 reports, originating from Germany, Russia, Brazil, Slovakia, Republic of France and 7 more countries
Gmail	Received 15 reports, originating from United States of America, Canada, Venezuela, Italy, Australia and 3 more countries
Verizon	Received reports, mostly originating from Dalton, Georgia, United States of America
Spotify	Received 11 reports, originating from United States of America, Republic of France, Canada, Brazil, Germany and 3 more countries
Nitrado	Received 4 reports, originating from Canada, United States of America, United Kingdom
Adobe Creative Cloud	Received 7 reports, originating from United States of America, Republic of Chile, Brazil, Germany, Republic of Austria and 1 more countries

COMPLEXITY

Q. Wow, all that failure must be expensive! Is there any way we can be more confident we are not going to fail?

A. Lets have controlled experiments where we try to identify weakness before it crashes the system.

COMPLEXITY

Sure, but 'controlled experiments' is a lame name. Lets call it 'Chaos Engineering'! Way cooler

COMPLEXITY

Chaos engineering is like a vaccine, which injects a small amount of virus to build immunity

CHAOS ENGINEERING

The harder it is to disrupt the steady state, the more confidence we have in the behavior of the system. If a weakness is uncovered, we now have a target for improvement before that behavior manifests in the system at large.

-https://principlesofchaos.org

CHAOS ENGINEERING – EXPECTED BENEFITS

Less downtime, better user experience

Less alarms and alerts (i.e. burnout) to Operations/SRE/Development teams

More productivity from less unplanned outages

Spreading knowledge of application to the team

CHAOS ENGINEERING

1. Define the normal/steady state of the system (monitor system and business metrics)

Hypothesis that steady state will continue in both control and experiment groups

2. Pseudo-randomly inject faults (kill containers, network, etc) simulating real world events

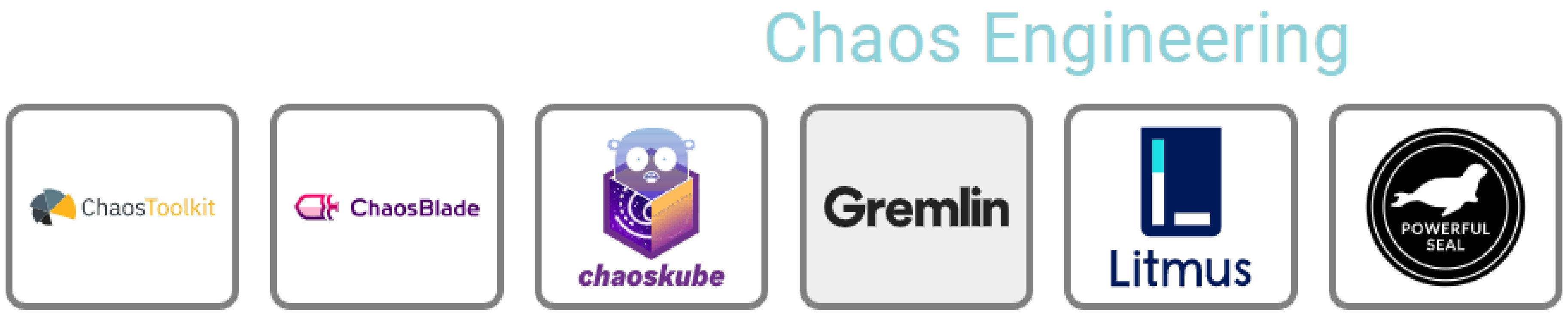
Try to disprove hypothesis looking for difference in control and experiment groups

GAMEDAY EXAMPLE – PER SERVICE

Failure Scenario	Experiment	Scoping	Signals/Metrics	Abort Conditions
SAMPLE: Application server latency	Latency	100ms - 1000ms – 2500ms	Service Availability, On call paging	SLA breach, RPS threshold reach

Expected Outcome	Actual Outcome	Bugs
Application should still be available, but slower	Total app failure	Fallback did not occur

CNCF LANDSCAPE



LIVE DEMO - SIMPLE CONTAINER EXPERIMENTS

PUMBA

Container Chaos experiment with Pumba

* Network delays!

LIVE DEMO – ADVANCED CONTAINER EXPERIMENTS

More advanced Chaos with Chaos Blade

* More Network Delays!

CHAOS ENGINEERING – SIMPLIFIED REVIEW

01

Have a hypothesis and identify control and experimental group

02

Use real-world events & limit scope

03

Make it as real as possible, ideally Production

04

Look for differences in steady state between control and experimental group

EXAMPLE EXPERIMENTS

Test

Test failure of a stack component to identify resiliency of individual components

Test

Test failure for an Availability Zone/Region (Rack if on premise) to identify group component failover resiliency

Test

Test slowdown of key services to identify dependency slowdown/intermittent failure

LIVE DEMO – CLUSTER EXPERIMENTS

Litmus

Test cluster failure with Litmus

ADDITIONAL TIPS

- Start Small
- Production if possible, or close to it
- Minimize blast radius
- Have an emergency stop

QUESTIONS?

Peter Lamar

VP Cloud & Developer Relations

Peter.Lamar@Softwareag.com

peterrlamar@gmail.com

@ptlamar