

*No-Nonsense*  
**Observability Improvement**

**Getting value, understanding cost, and learning.**

# Hello, I'm Cory!

---

- **Splunk:** Technical Evangelist?

*Previously*

- **SignalFx:** Director, Office of the CTO
- **Stripe:** Principal Engineer, Observability Lead @ Stripe
- **Twitter:** SRE & Engineering Manager

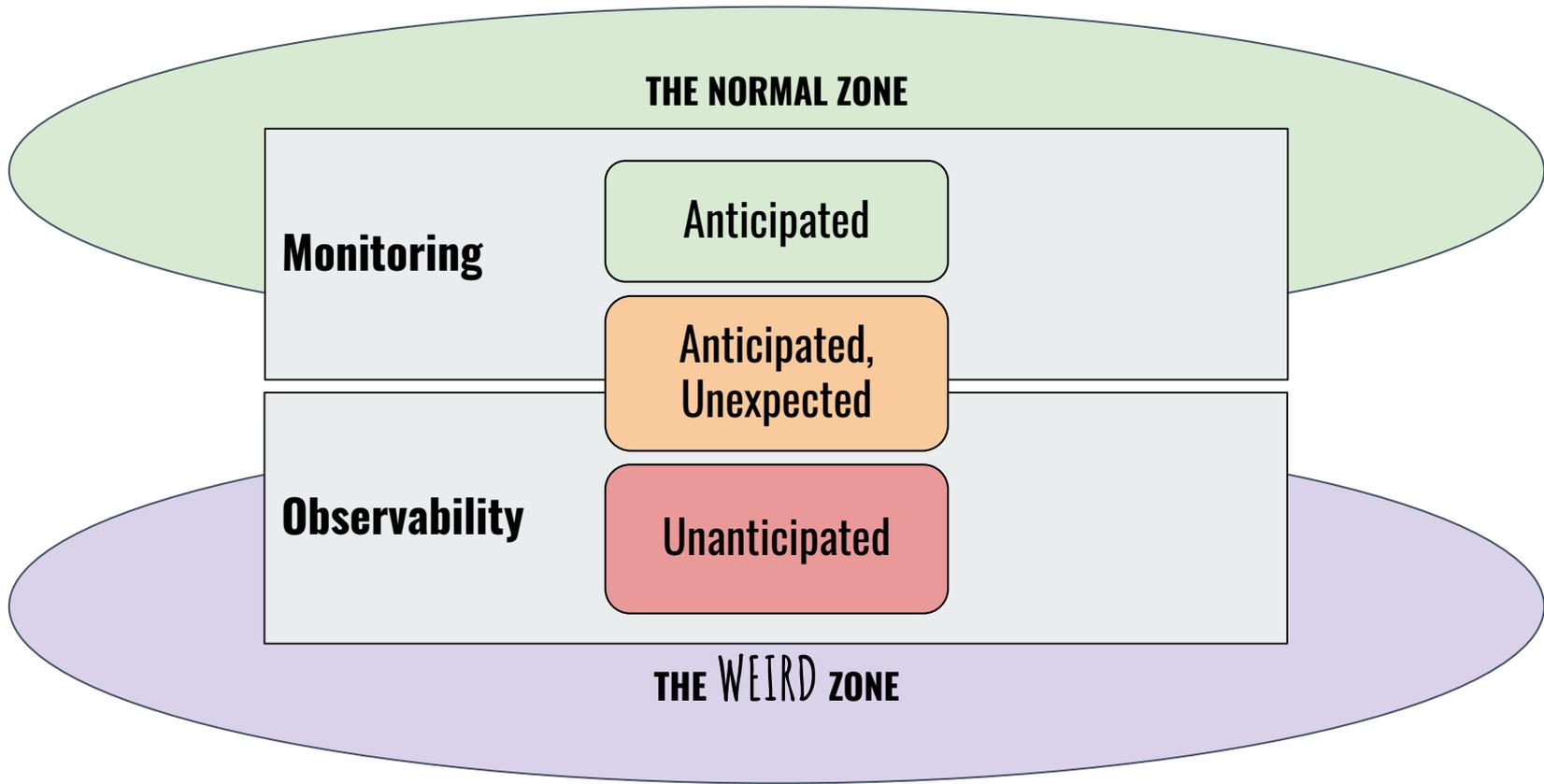
onemogin.com, @gphat



# *Today's Goal*

Useful, vendor agnostic ways to improve.

**Why do we observe?**

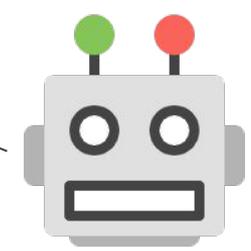
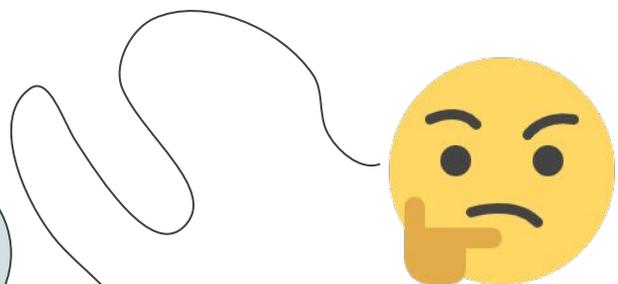
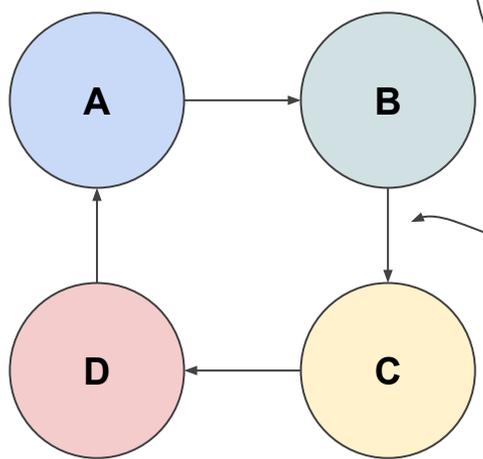


Observability, Visualized

**Systems are complex.**

**Systems are complex,  
non-linear, chaotic,  
and emergent.**

# Our System



**Observability?**

*No-Nonsense #1*  
Promote Your Work

*No-Nonsense #2*  
**Teach Basic Skills**

Your first interactions  
with tooling should  
**not** be under duress  
at 3am.

# Basic Skills

— — —

- Humans learn by doing, not by watching or being told
- But we can get them started!
  - Navigation
  - Finding metrics
  - Any common frameworks / approaches
  - Making dashboards, alerts, etc
  - Plug your stuff!
- What do you need to know to get started on your own?

# Practice!

— — —

- More things to do!
- What are the next steps?
- What can you do week over week?
- Martial arts as an example

*No-Nonsense #3*  
**Stretch Your Systems**

Improvements in  
capacity are  
consumed by  
productivity and  
efficiency.

# Do the stretching yourself

— — —

- Not outside influences
- Test improvement
- Resilience engineering
  - Load testing
  - Chaos testing
  - Gamedays
- Incident flags

*No-Nonsense #4*  
Measure Usage

**Observability will be  
one of your most  
expensive projects.**

# Measuring (Write) Usage

— — —

- Infra spend?
- Dependencies
- Service association
- Team association
- Changes over time
- Consumption

# Measuring (Read) Usage

— — —

- Monitoring
- Sources used in investigations
  - Incidents...
  - Chaos
  - Gamedays Etc
- Dashboards
  - More usage!
- Keep going!

*No-Nonsense #5*  
Track Need In Detail

**Complex systems are  
hazardous, change a  
lot, and require  
experience.**

# Incident Measures

— — —

- Traditional measures are pretty lame
- MTTD, MTTR, timestamp, duration, recipient, severity, “root cause”
- Not enough information

# Incident Measures++

— — —

- Credit: Nora Jones (@nora\_js)
- Difficulties in understanding
- System-specific failure rates (and notably, those that **haven't** failed)
- Surprises
- Lack of ownership
- Uncommon pairings of teams/services
- Near misses
- **Feed back to stretching!**

*No-Nonsense #6*  
Instrument Change

The passage of time  
is not the primary  
mutator in systems.

# Humans (and Robots) Change Things

— — —

- Deploys are obvious
- Change management, feature flags, scale up/down
- Automation
  - Gives up at bad times
  - Need to avoid human being “out of the loop”

# Give Feedback

— — —

- Get info to the user
- Toggled on/off easily
- Avoid purely “analytic” interfaces
- Outside the box: alerts? Other signals?

*No-Nonsense*

**Observability  
Improvement**

- Promote the work
- Teach basics
- Stretch systems
- Measure usage
- Track need
- Instrument change

---

How does this  
improve  
observability?

# Promote The Work

- Awareness
- Stygmergy
- No press is bad press
- Lean on leadership to help



# Teach Basics

- Get 'em familiar
- Establish “your way”
- Avoid frustration



# Stretch Systems

- Leverage data to generate improvement
- Wring more value out of the data
- More resilience and reliability
- Better understanding through practice



# Measure Usage

- Know the cost
- Invest in risk and need
- Dial back some stuff
- Be prepared when finance starts asking



# Track Need

- Improve your measure of incidents
- Examine success as often as failure
- Generate more questions for stretching



# Instrument Change

- Change is often the start of problems
- Eliminating change as the source is good too
- Quicker time to clue
- More change is good!



Some of all of these  
should mean a more  
rigorous, valuable  
observability  
program.

# Thanks!

@gphat, onemogin.com

# Citations

— — —

- Vicente & Rasmussen (1992) Ecological Interface Design: Theoretical Foundations. IEEE Transactions on Systems, Man, and Cybernetics, Vol 22, No. 4
- Jones, Nora (2019) [Before, During and After Chaos - Creating Foresight through a Cyclic Approach](#). Monitorama Baltimore
- Bainbridge, Lisanne (1983) Ironies of Automation. Automatica, Vol 19. No 6, pp 775-779
- Woods DD, Cook RI (2002) Nine Steps to Move Forward from Error. Cognition, Technology & Work
- Wikipedia contributors (2019) Complex system [Internet]. Wikipedia, The Free Encyclopedia