

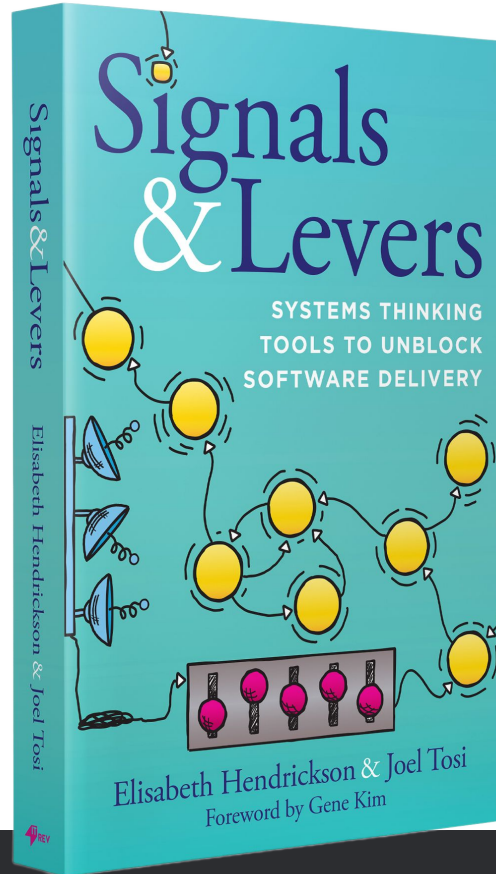
Poking Systems

Using Systems Thinking to
Navigate Non-Linear Complexity

Elisabeth Hendrickson
CAST 2026

this material is from the book...

Elisabeth
Hendrickson



Joel Tosi



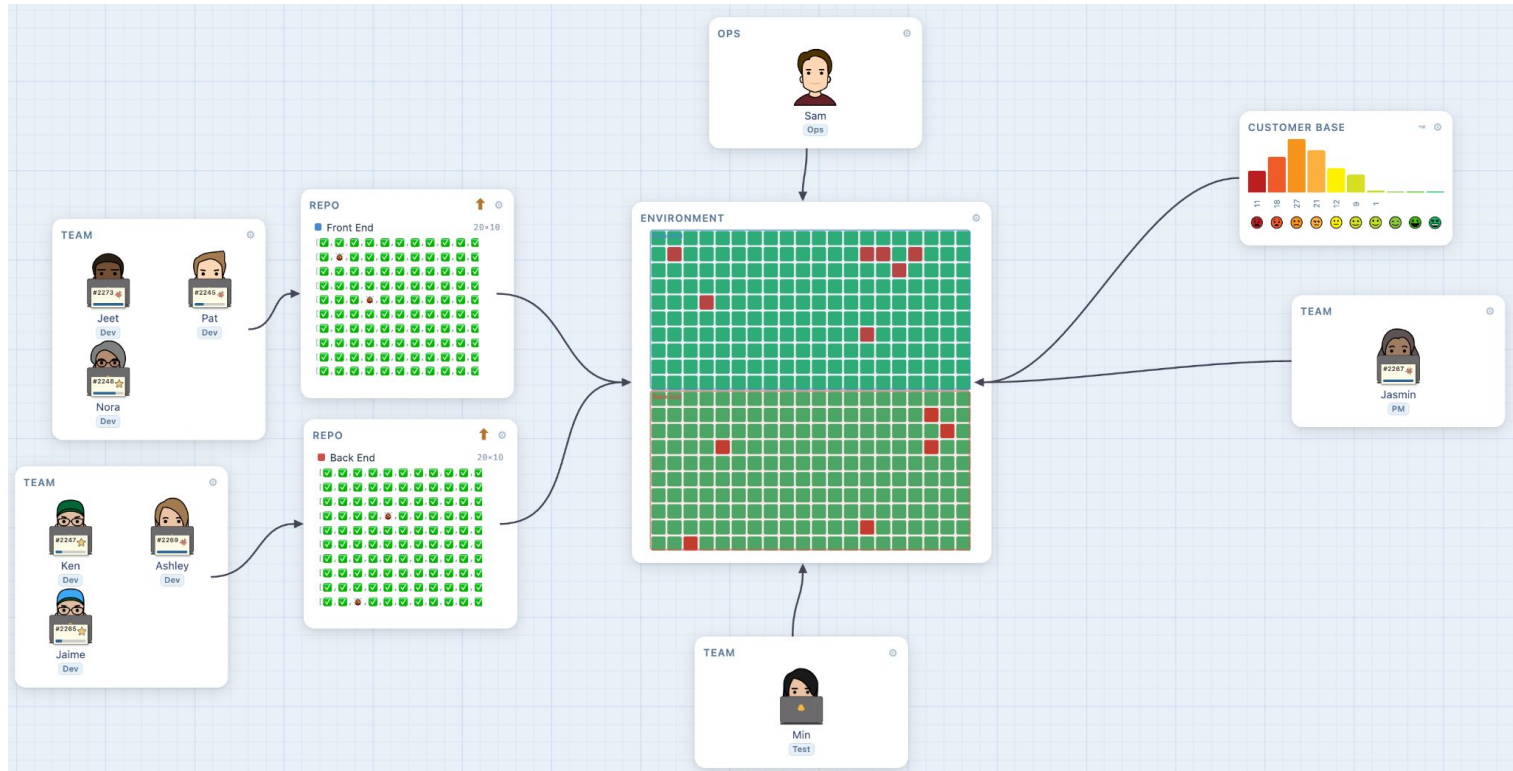
slides +
other
resources



Agenda

1. Overview: Signals, Levers, and Systems
2. Detecting Signals
3. Visualizing Influences with Causal Models
4. Seeing Time Delays
5. Finding Levers
6. Experiencing Unintended Consequences
7. Wrap Up

With exercises using a simulation...



explore

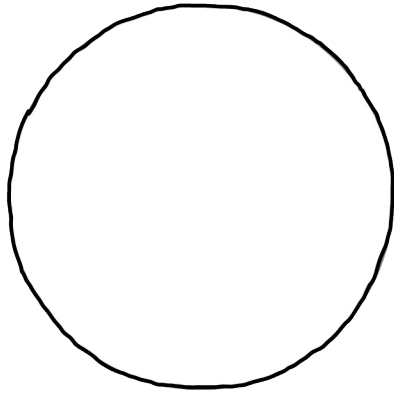
introducing the simulation
sim.signalsandlevers.com

Overview

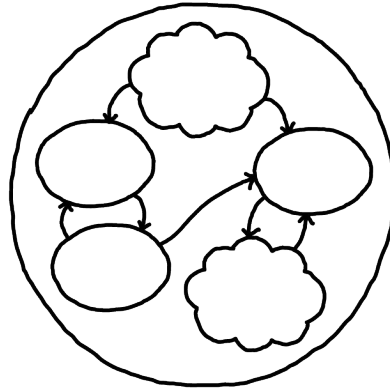
signals, levers, & systems

About Systems...

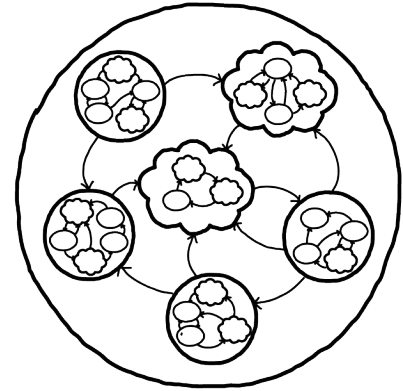
A **system** is a set of interrelated parts that does something. It has these properties:



the whole is
indivisible

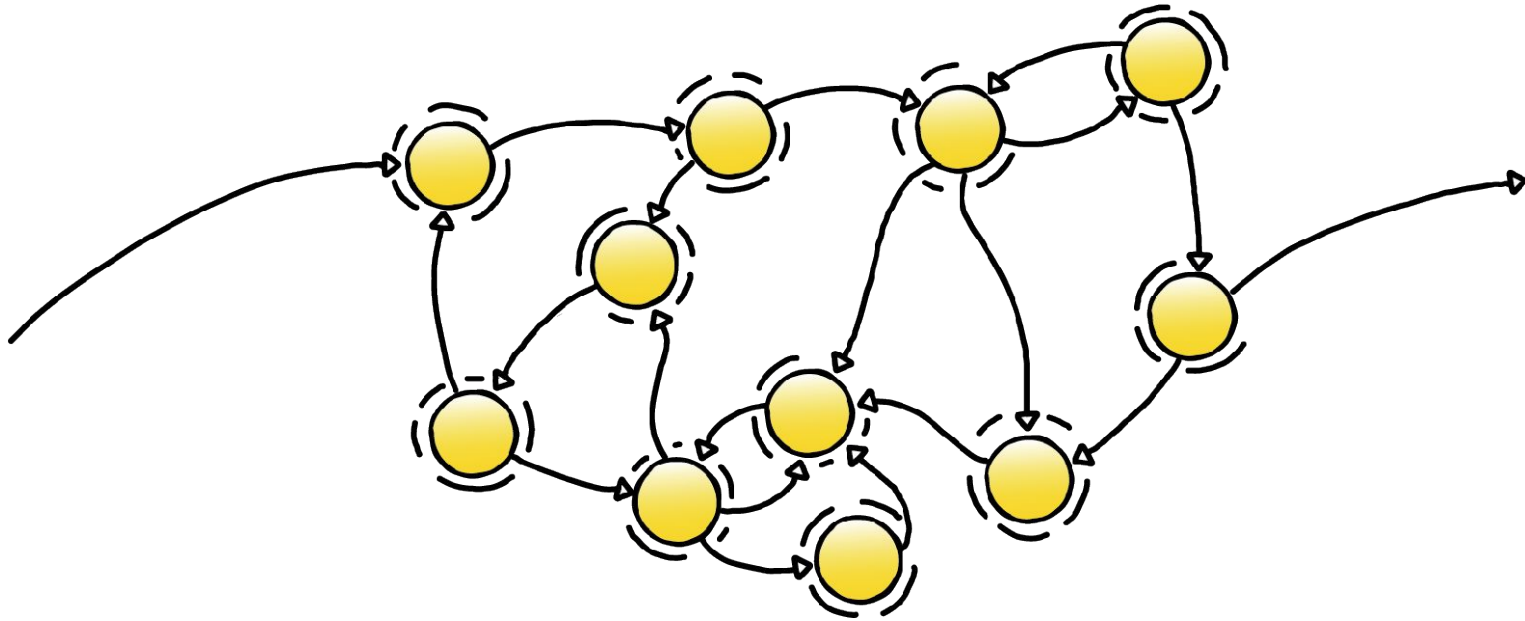


parts are
interdependent



systems are
fractal

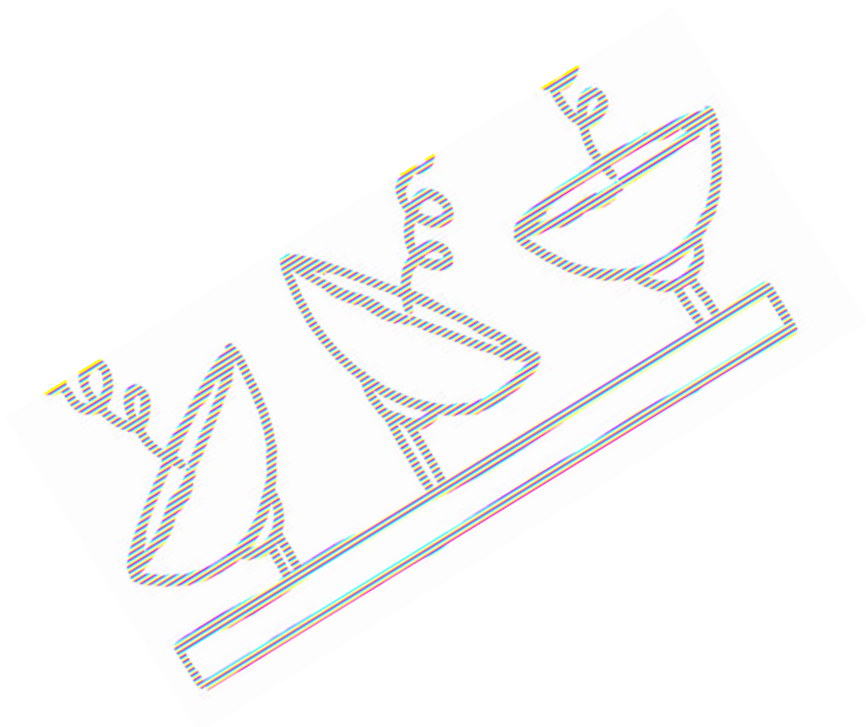
Software Delivery is an Adaptive Sociotechnical System



Signals

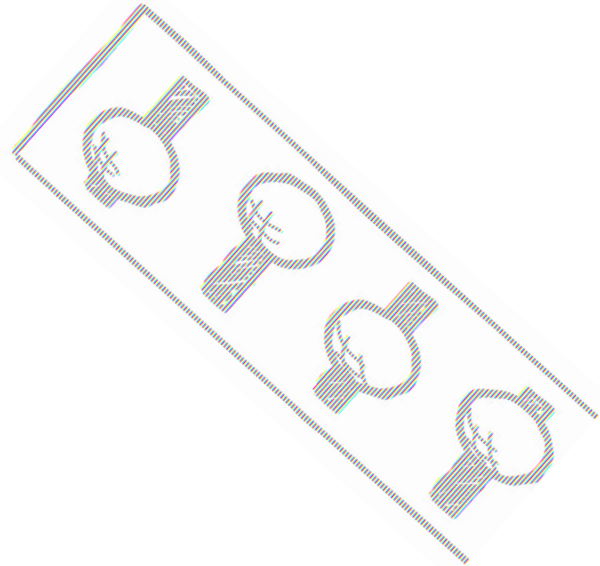
Signals are observable characteristics or attributes of the system that tell you something about the way the system works.

Noise is irrelevant, misleading, or distracting data with no correlation to outcomes.



Levers

Levers are choices or decisions you can make to cause a shift in the system.



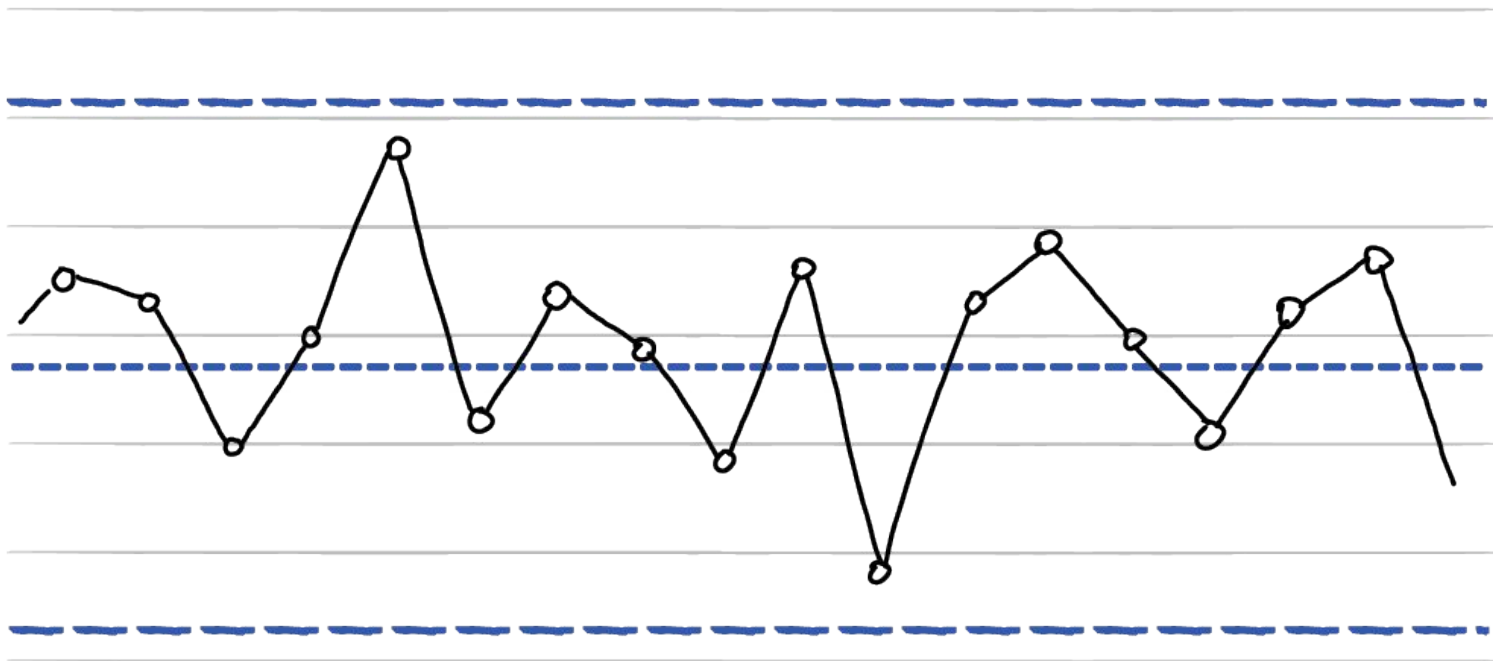
Detecting Signals

the things you see & hear that
inform you about the system

Metrics are signals

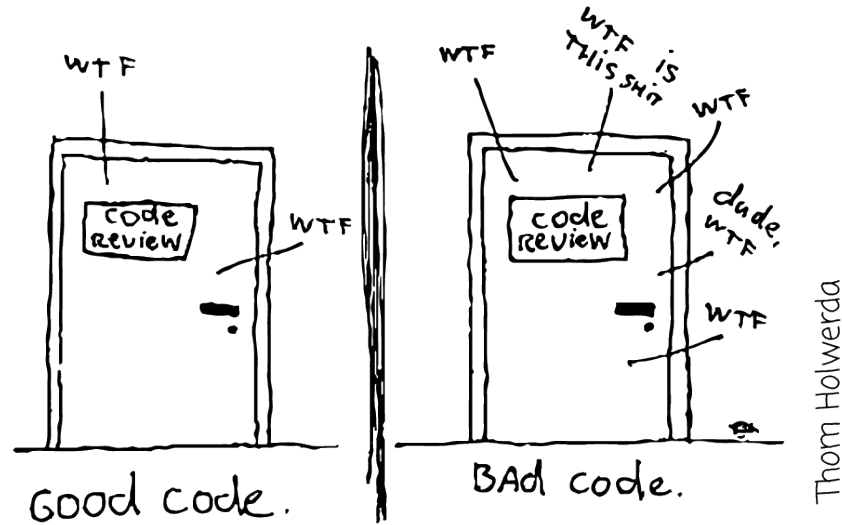


Variability is a signal



Interactions are signals

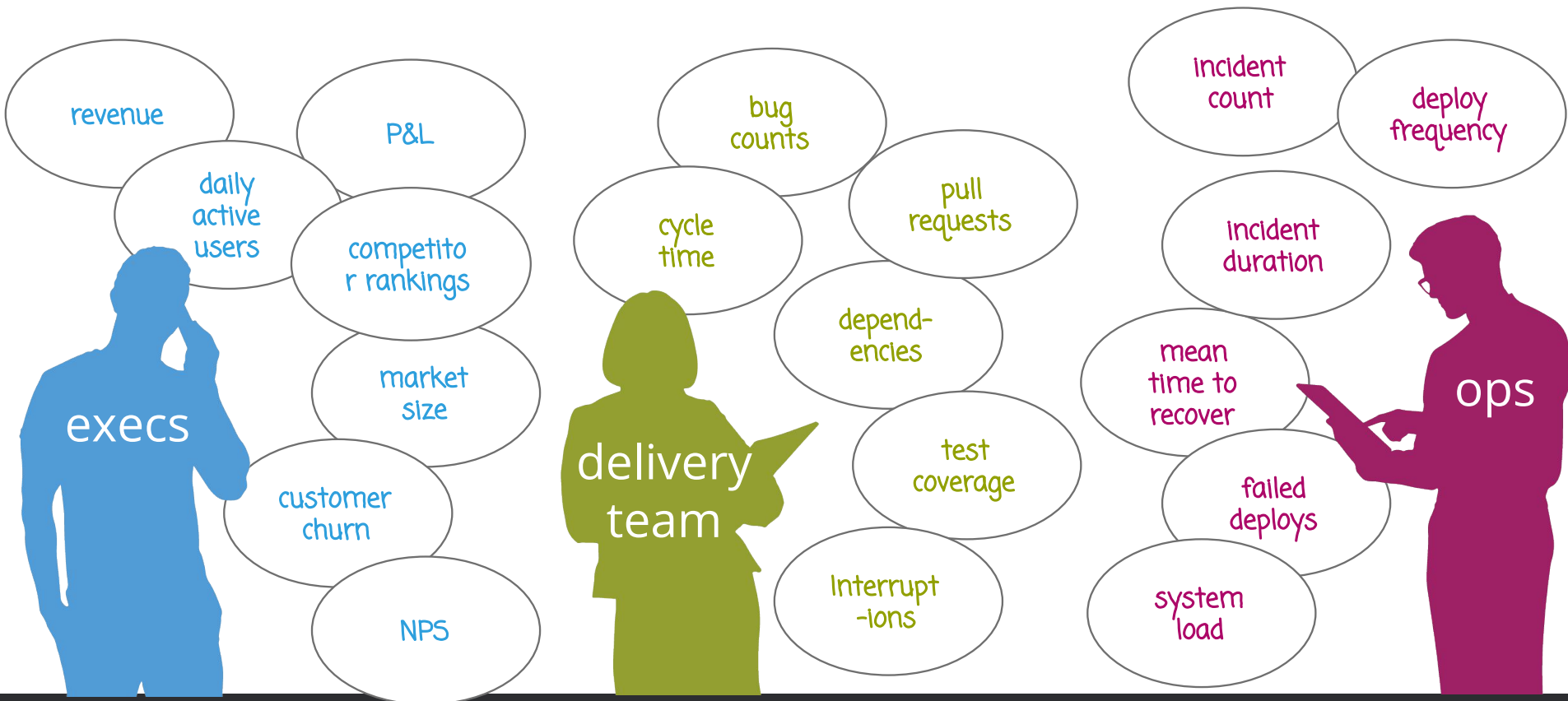
The ONLY VALID MEASUREMENT
OF CODE QUALITY: WTFs/MINUTE



Cartoon by Thom Holwerda. Original: <https://www.osnews.com/story/19266/wtfs/>

You are surrounded
by signals.

Different roles see different signals



What signals do you use to
figure out what's happening
inside your system?

explore

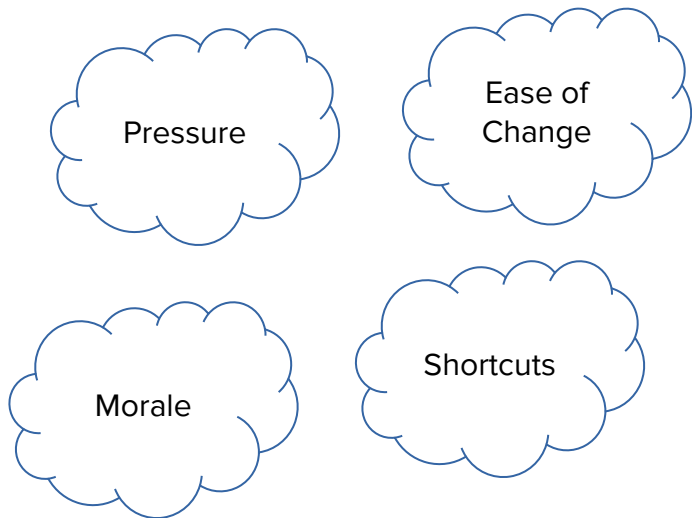
seeing signals in the simulation

Visualizing Influences

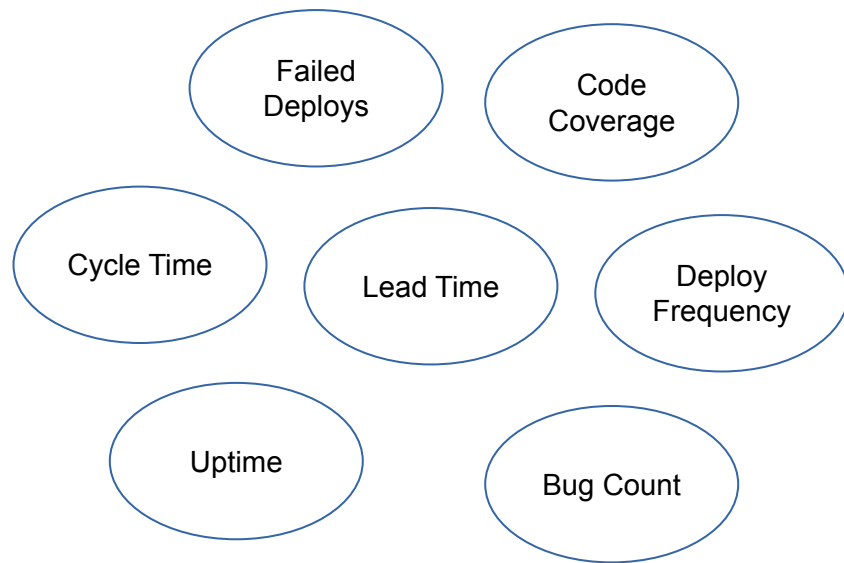
how to express relationships between
system attributes with causal models

Qualitative & quantifiable attributes are both valuable

qualitative



quantifiable



Examples of attributes

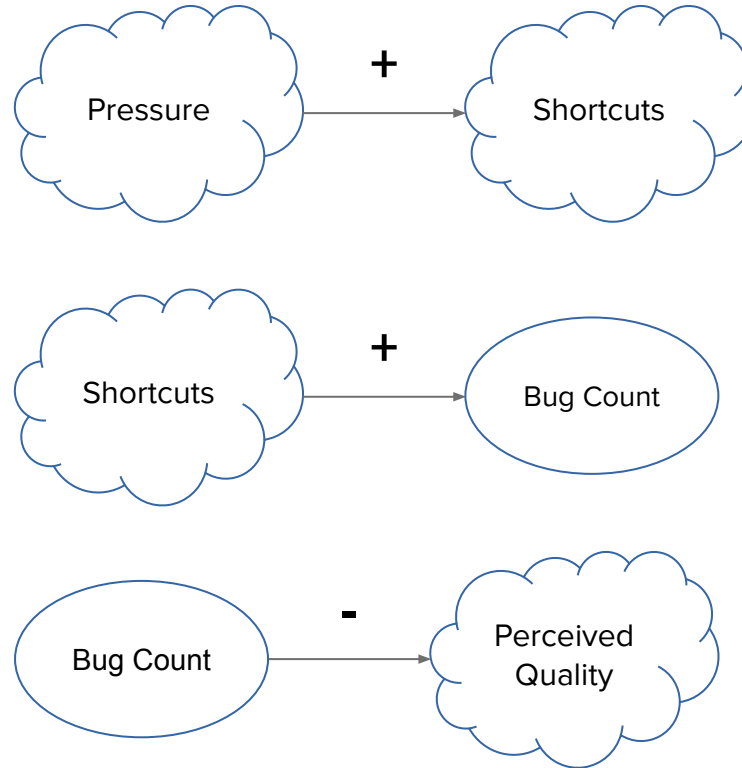
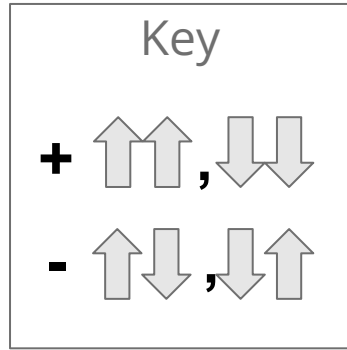
Qualitative

- Pressure
- Shortcuts
- Quality
- Stability
- Churn
- Risk
- Ease
- Understanding
- Alignment
- Innovation
- Cognitive load
- Technical debt

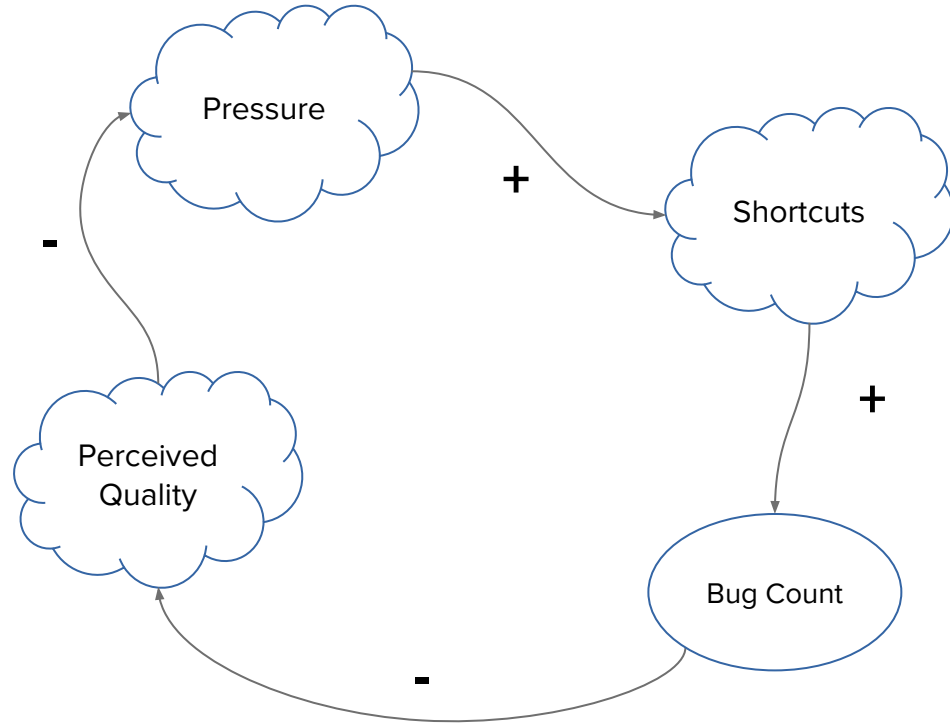
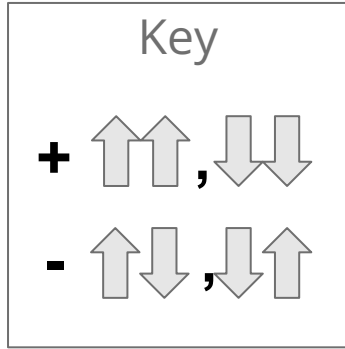
Quantifiable

- Cards / tickets
- Bug metrics
- Code metrics
- Test pass / fail
- Velocity
- Time (elapsed, effort, waiting)
- Production incidents
- Failed deploys / rollbacks
- Queue depth
- WIP
- Support cases
- Interruptions

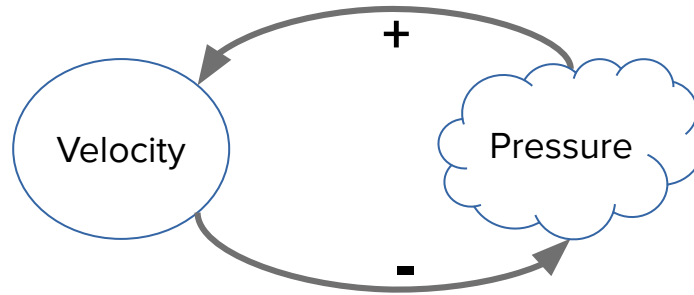
Attributes influence one another



Loops make systems adaptive

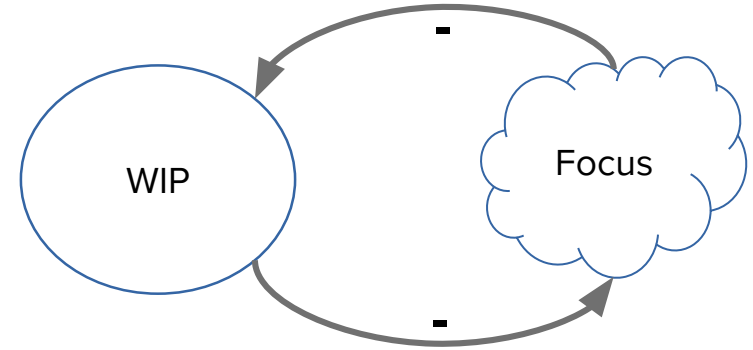
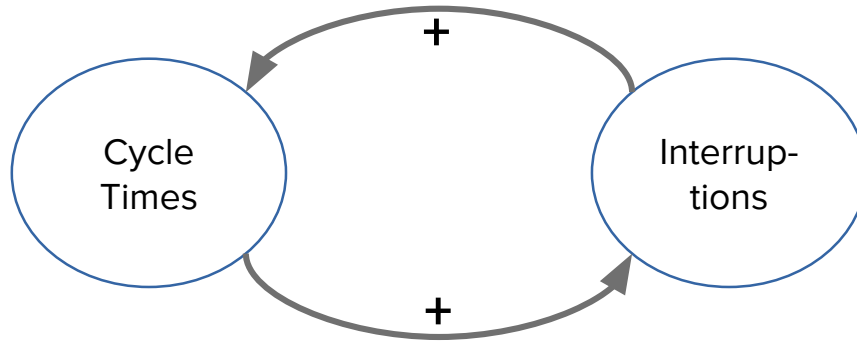


Balancing loops



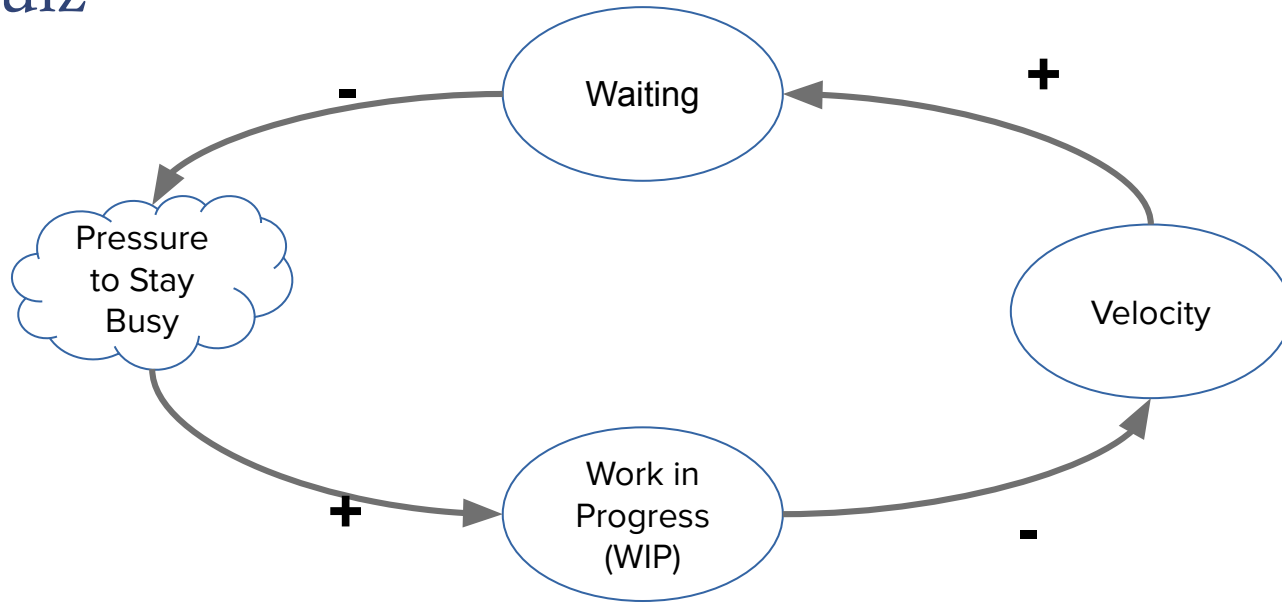
Self-Correcting
Dampening
Stabilizing

Reinforcing loops



Continued Growth
Can be "virtuous" or "vicious"
Can lead to instability
Amplifying

Pop quiz



Balancing or Reinforcing?

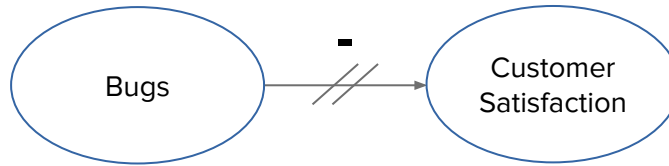
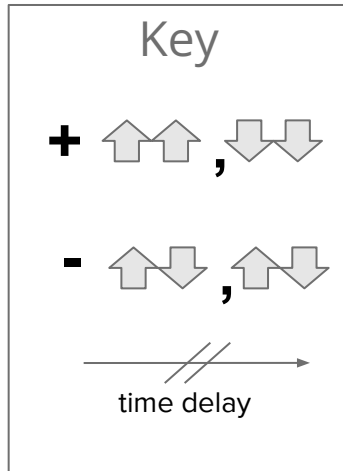
explore

modeling influences in the simulation

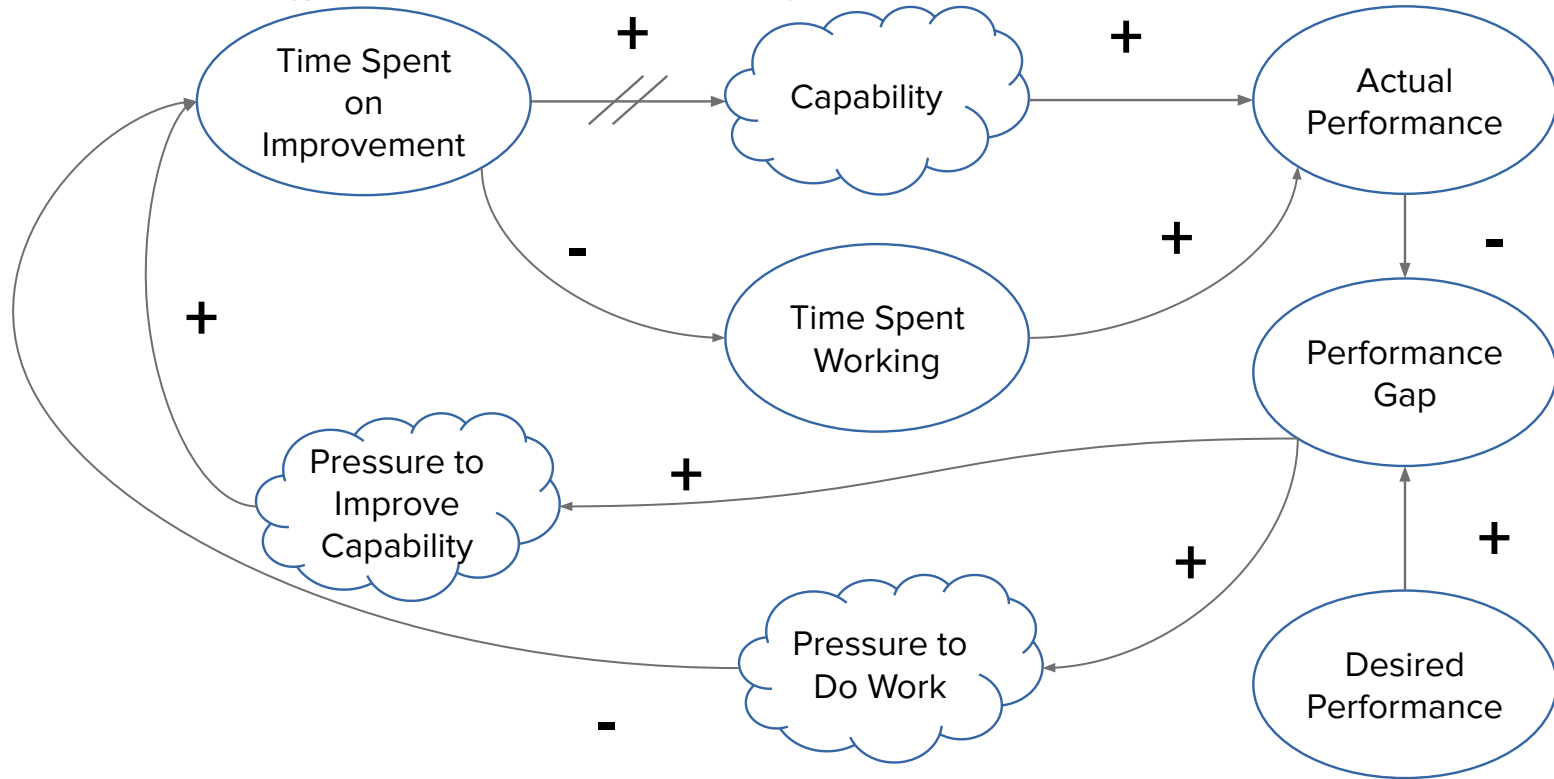
Seeing Time Delays

feedback latency makes it
harder to read signals

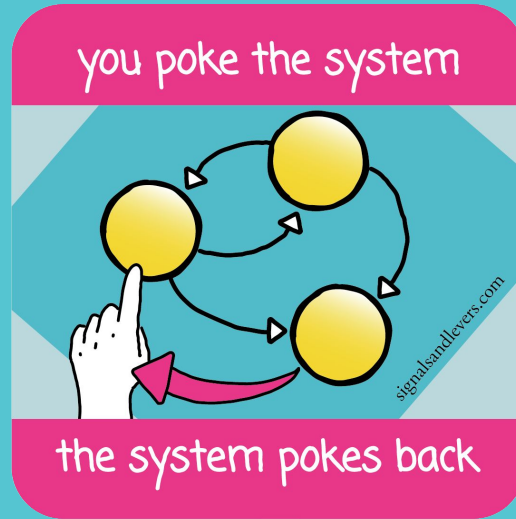
Showing time delay



An example of time delay



SURPRISE!



Common examples of delays

- Planning Horizons
- Multiple Step 'Process'
- Applying New Skills
- Skill Isolation / Dependencies
- Understanding New Work
- Work Spanning Time Zones
- Customer Feedback
- Separating Learning from Doing
- Support Tiers
- Team Bonding
- Quality Gates (External QA, PR Reviews)
- Trust

Which ones can you directly influence?

explore

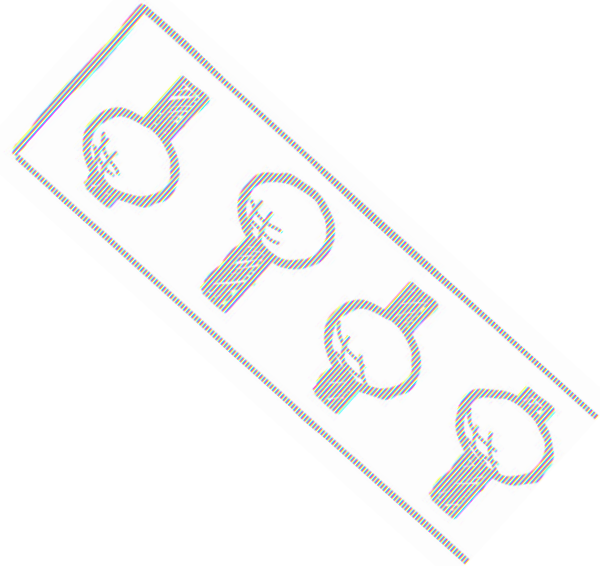
increasing and decreasing
time delays in the simulation

Finding Levers

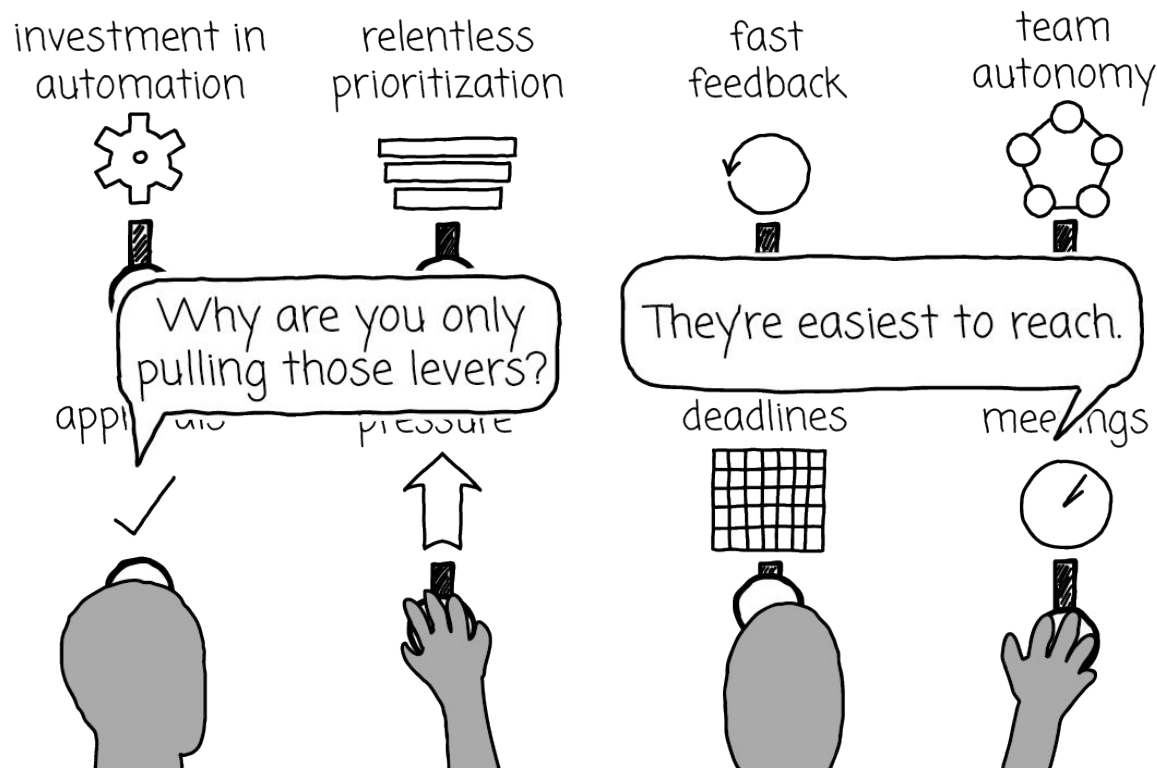
how to identify decision points

Levers

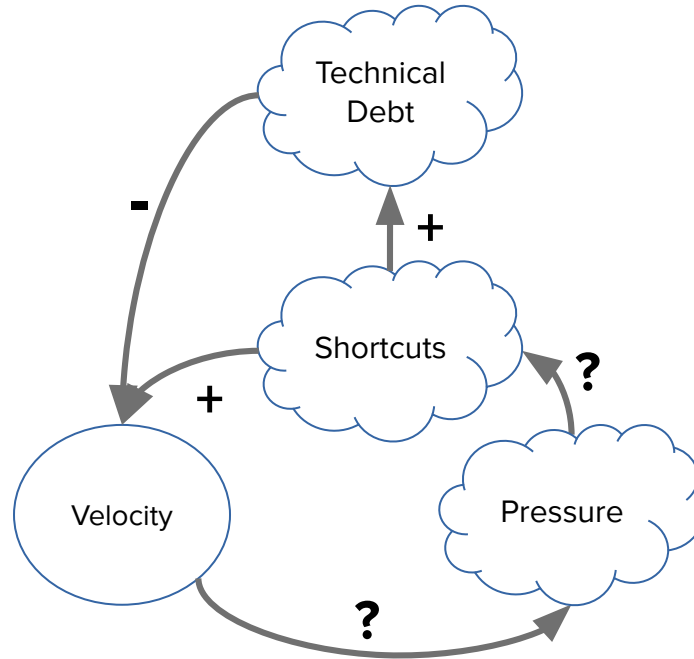
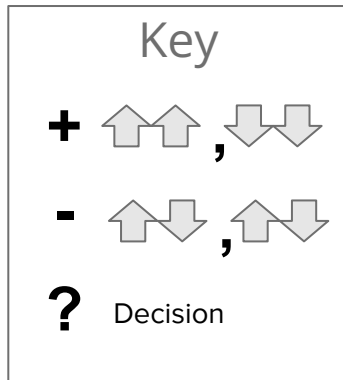
Levers are choices or decisions you can make to cause a shift in the system.



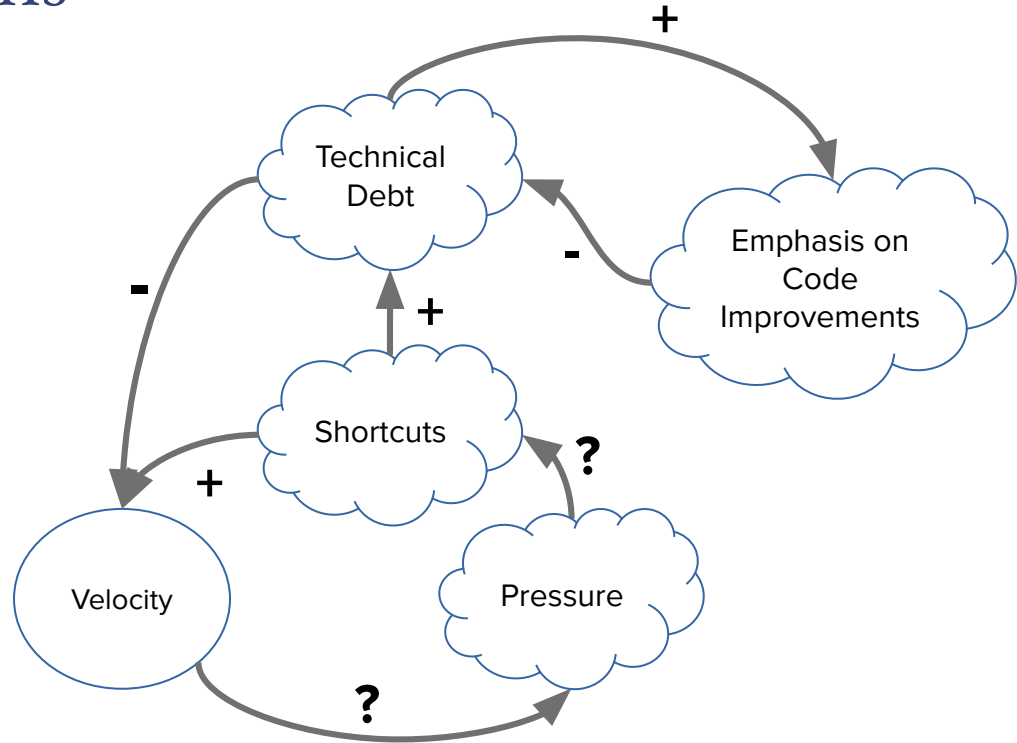
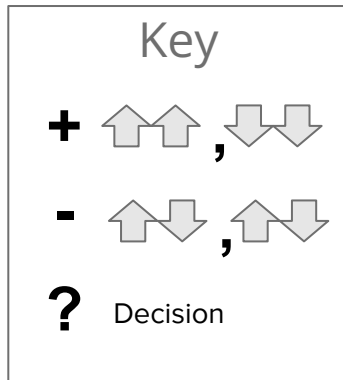
The easiest levers usually aren't the most effective



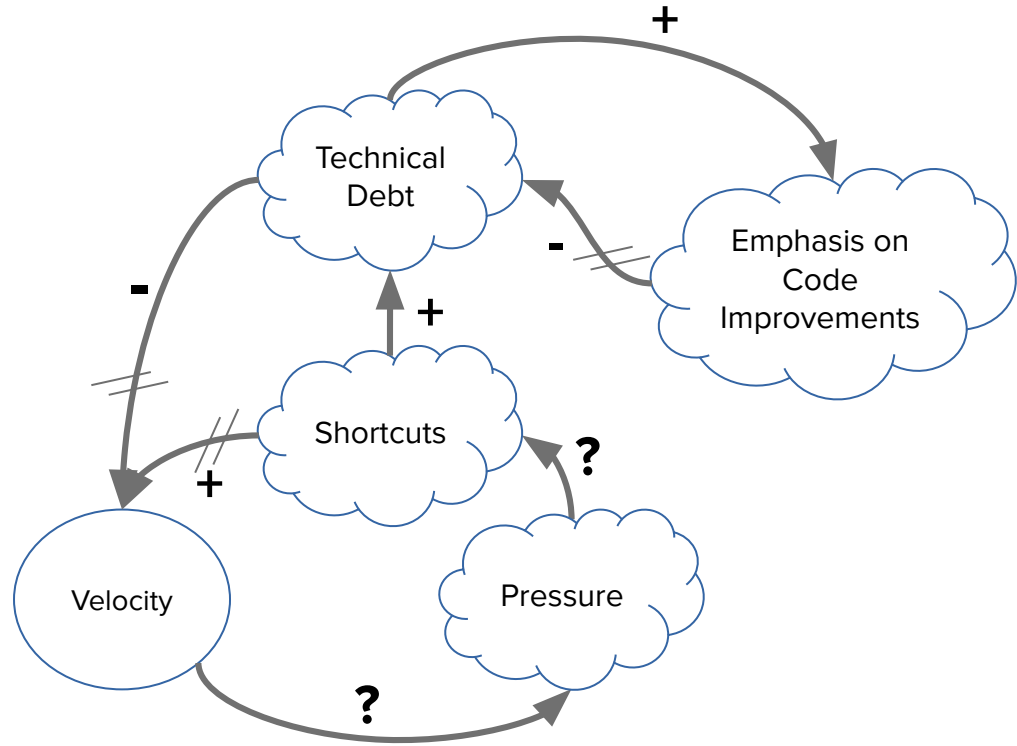
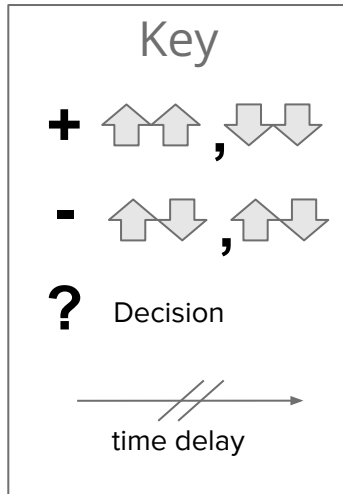
Finding choices



Finding interventions



Time delay makes good choices harder





Flying Buttresses

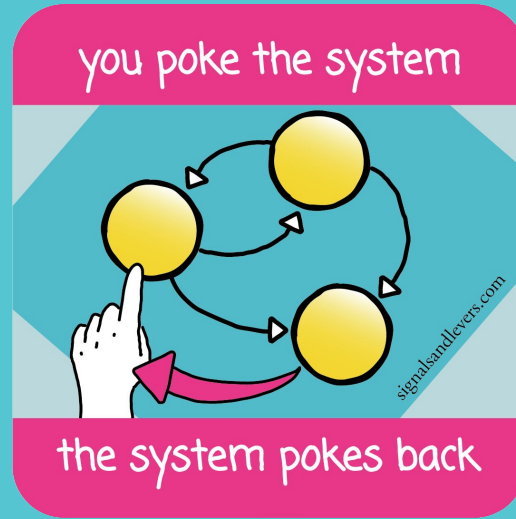


A **flying buttress** is a process or policy designed to shore up another process or policy that can't hold on its own.

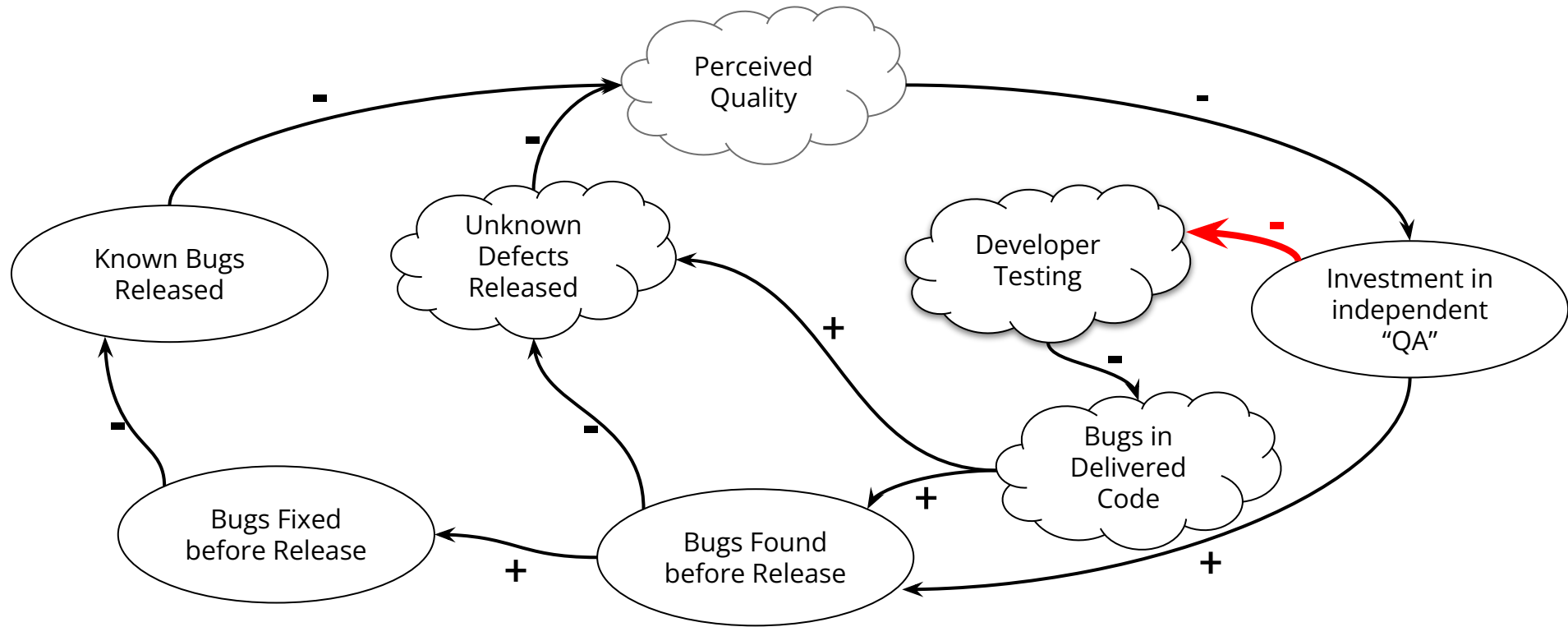
Examples:

- Pre- pre- meetings
- A formal expediting process to bypass process overhead
- Overstandardization

beware unintended consequences

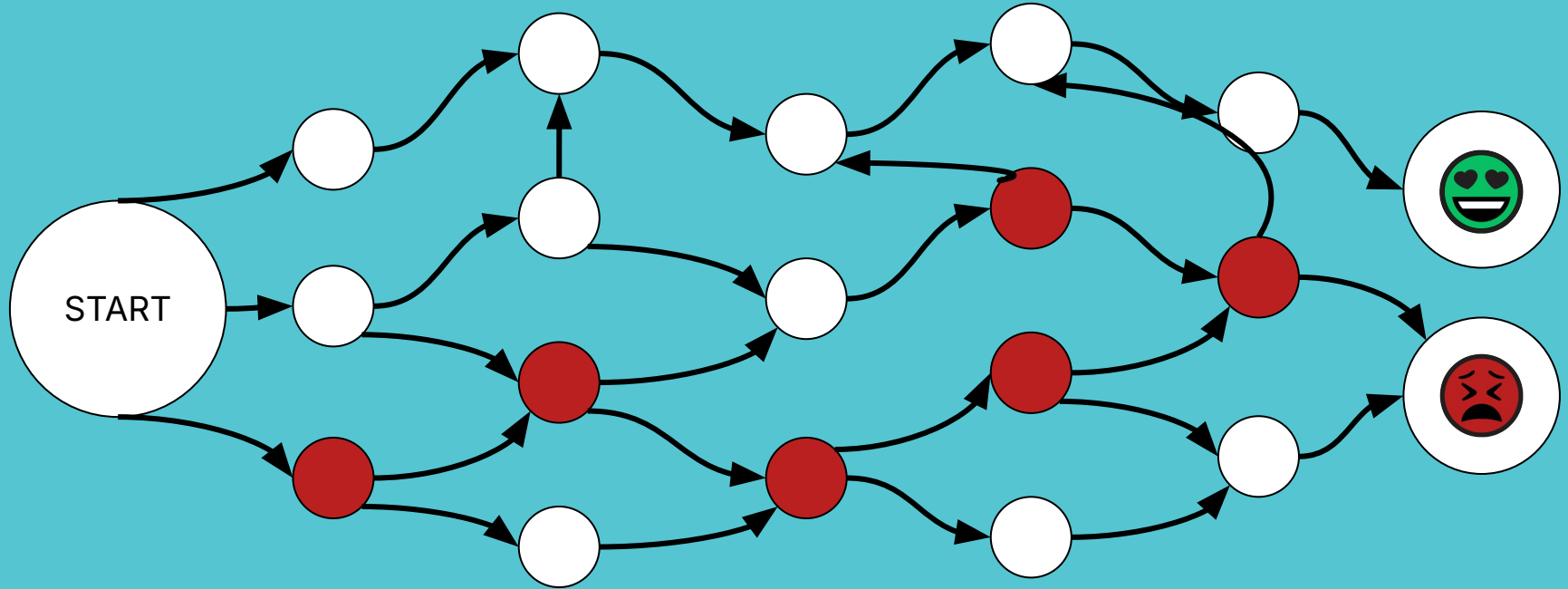


Better testing, worse quality



Act where pain is caused

not where it is observed

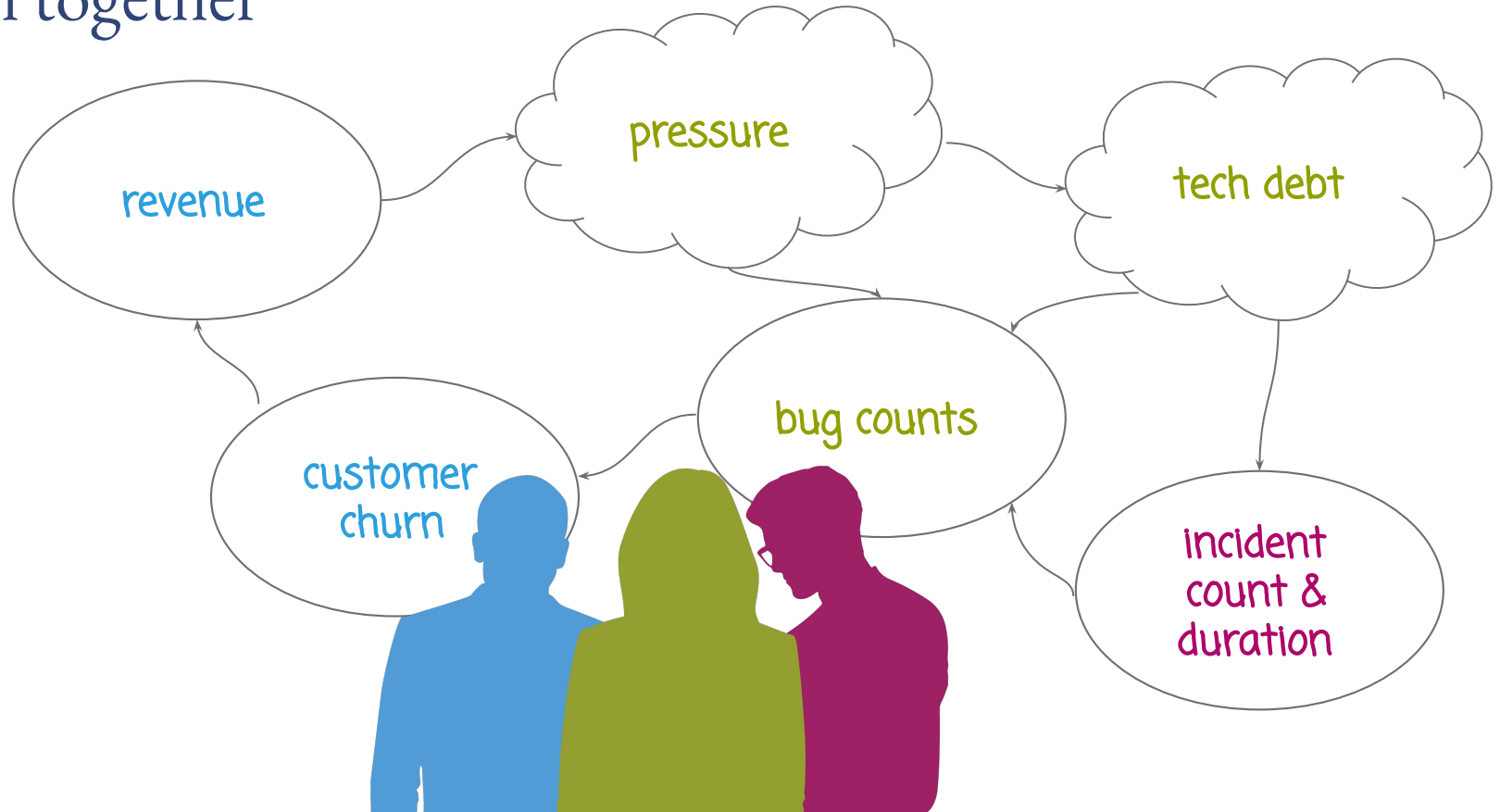


explore
finding surprises

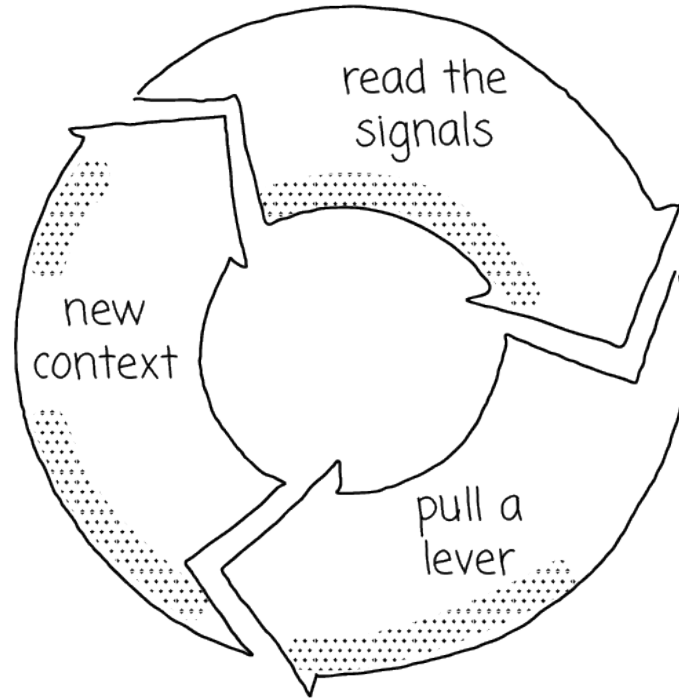
Wrap Up

final thoughts & next steps

Model together



Your context is constantly changing



levers operate in context

Right +  = Wrong

Wrong +  = Right

discuss

What will you do differently next week?

Thank You!

slides + other
resources

