

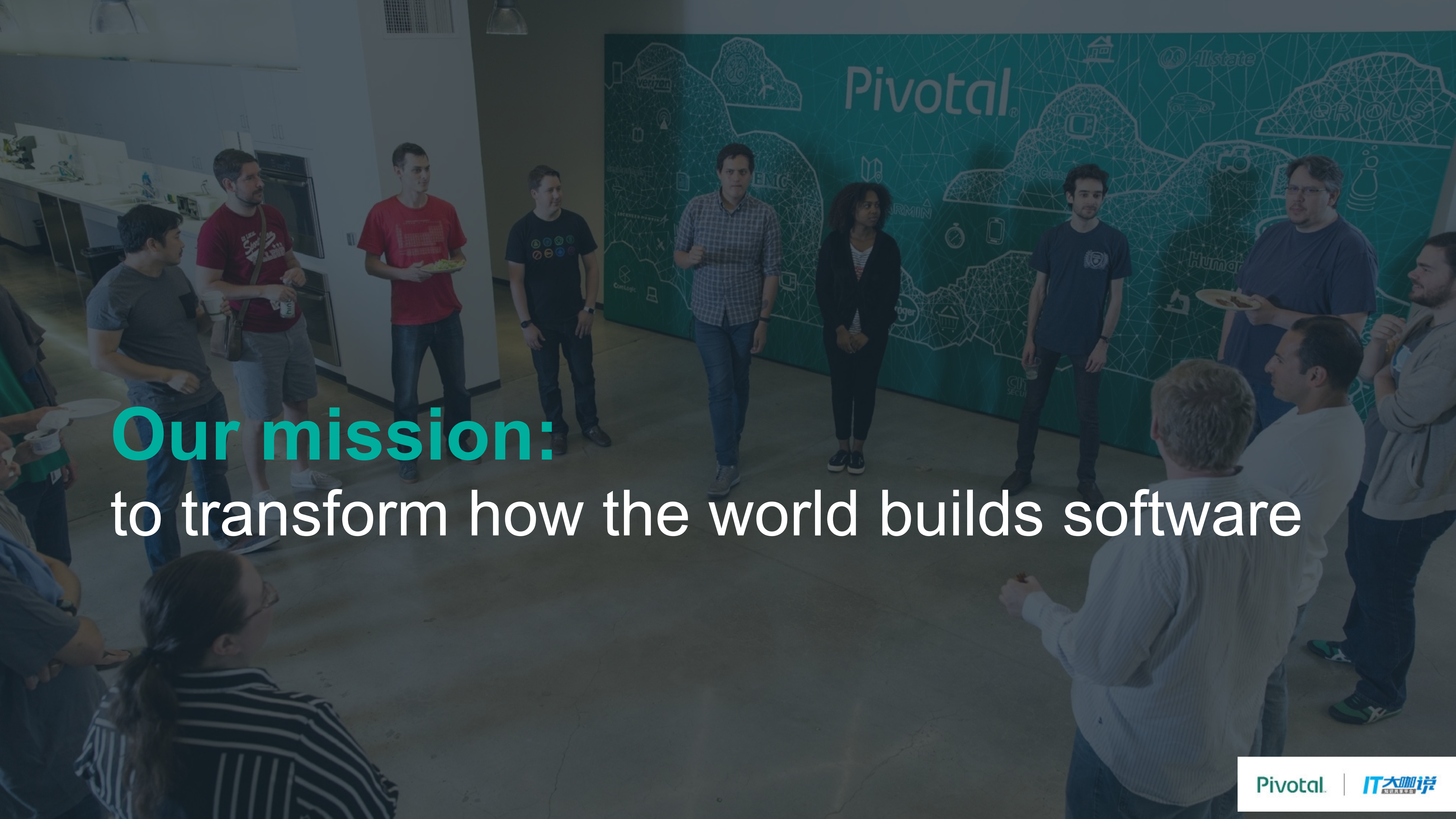
# elemental devops

Andrew Clay Shafer  
@littleidea



Pivotal™

北京



**Our mission:**  
to transform how the world builds software

## Why do you need to be good at software?

**Customers  
expect it.**

**Meet the  
demands to  
operate at  
scale.**

**Give you  
more  
business  
options.**

**Your  
competitors  
are  
improving.**

**It makes your  
life better.**

# What keeps you from being good at software?

**It's hard to experiment and quickly incorporate what you learn.**

**Stuck with incomplete or outdated application platforms.**

**Hostile processes and procedures make it painful to ship software.**

**Organization silos have competing priorities.**

## The transformation is real.



T-Mobile goes from **7 months and 72 steps** to update software, to **same day deployments**.



Liberty Mutual **builds and deploys an MVP in one month** and delivers revenue-generating version just hours later.



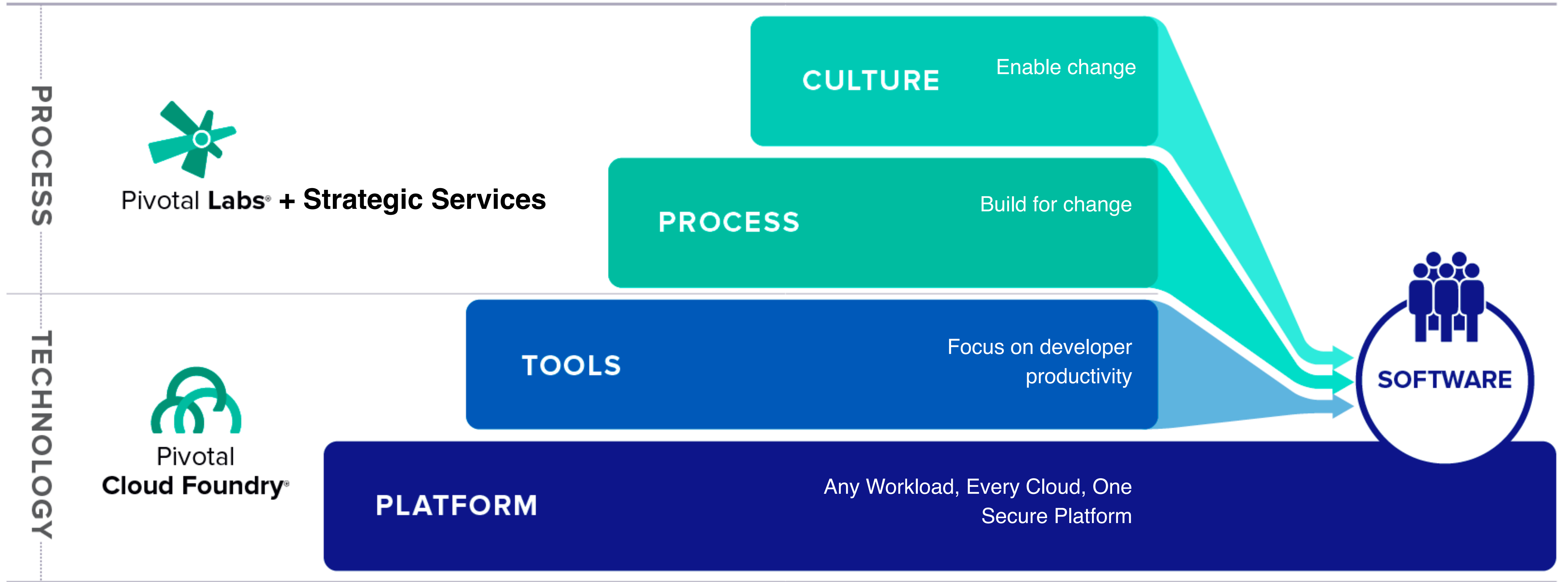
The Home Depot **ships to production 1,500 times a month**, and 17,000 times a month to all environments.



Comcast supports over 1500 developers with an **operator team of 4 people**.



Express Scripts went from **45 days** to patch one product in nine environments, to **five days**.





**My mission:**  
to transform how the world operates software

# Andrew Clay Shafer

---



# Pivotal™





# Andrew Clay Shafer

---

@littleidea

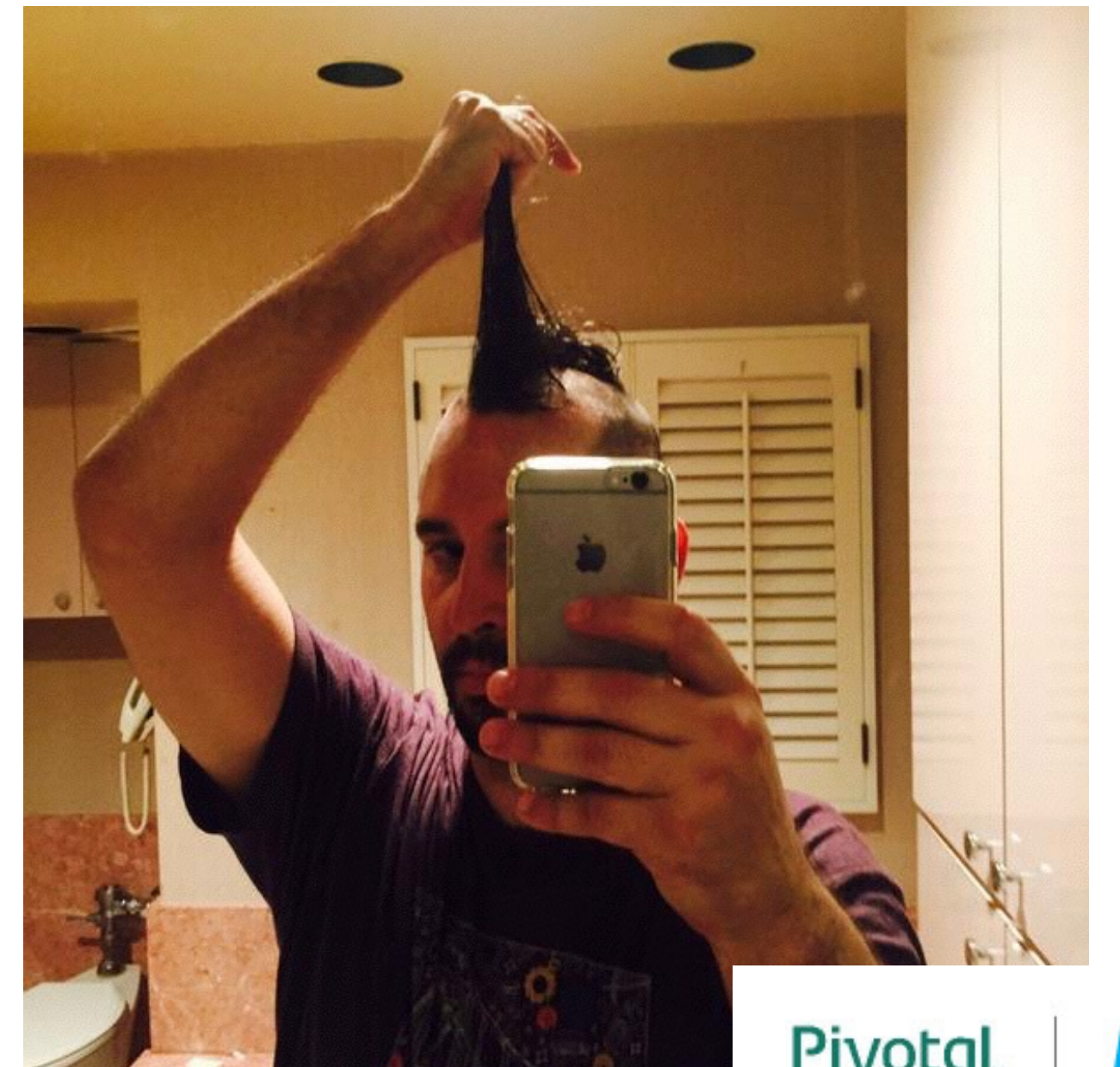
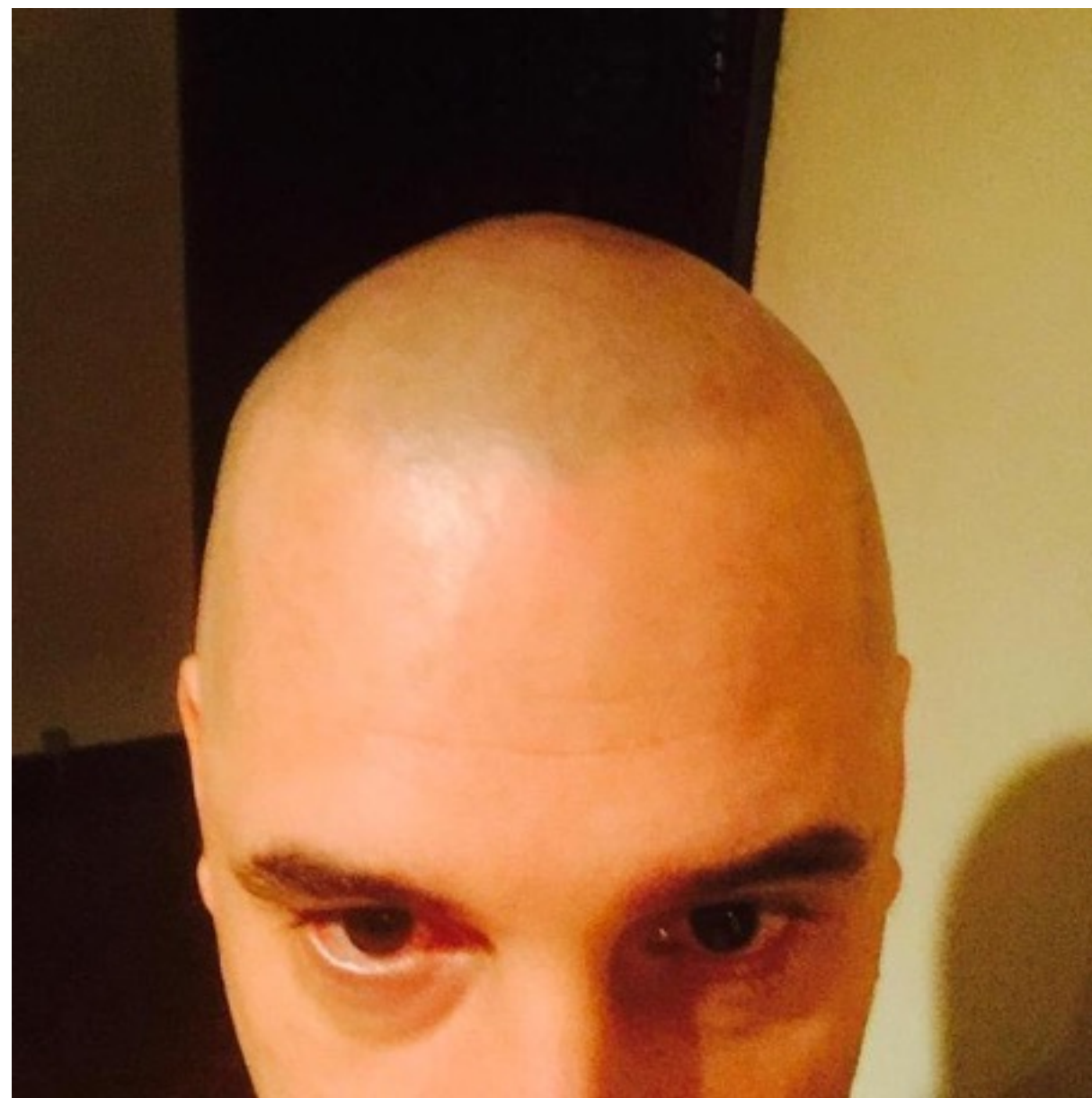


Andrew Clay Shafer

---

@littleidea







# spring







# most important devops stuff

- Learn to Read
- Learn to Write
- Learn to Speak
- Follow [@littleidea](#) on Twitter



tl;dr

the way to improve dev...

is improve ops...



tl;dr

the way to improve ops...

is improve dev...

but why?

who cares?

One Word...

SOFTWARE

tl;dr

you are building a software company

or losing to someone who is

# tl;dr

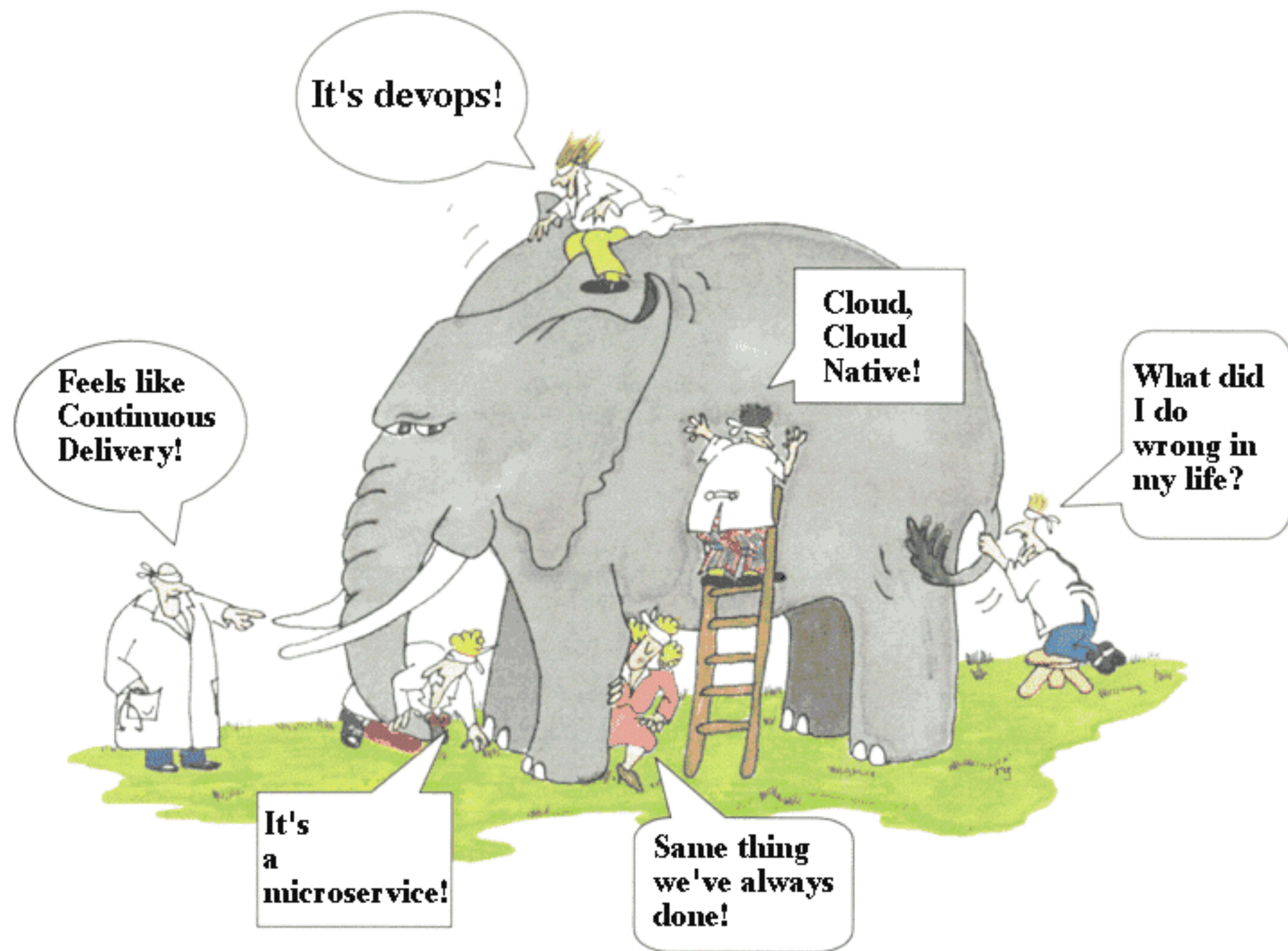
you are continuously devopsing microservices

or losing to someone who is

continuous delivery,  
devops,  
microservices...

# These things are all one...

---





tl;dr

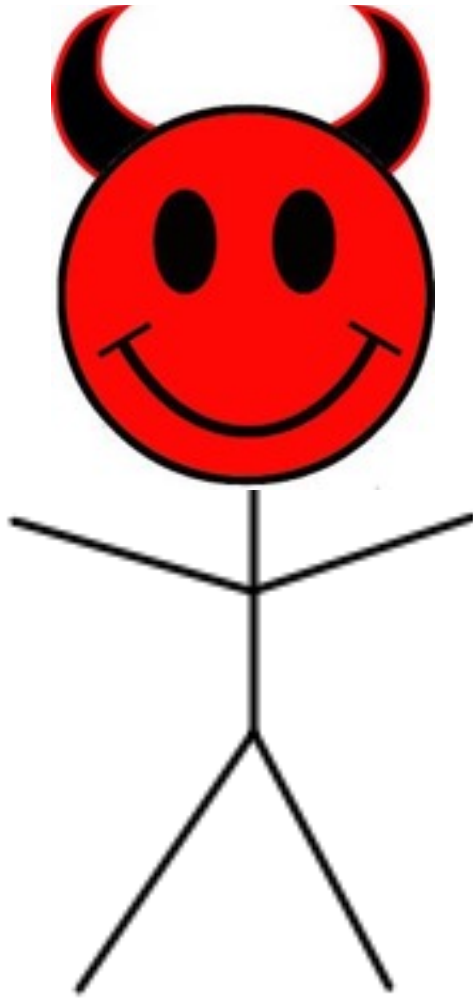
you are building a learning organization  
or losing to someone who is

learning is sustainable advantage

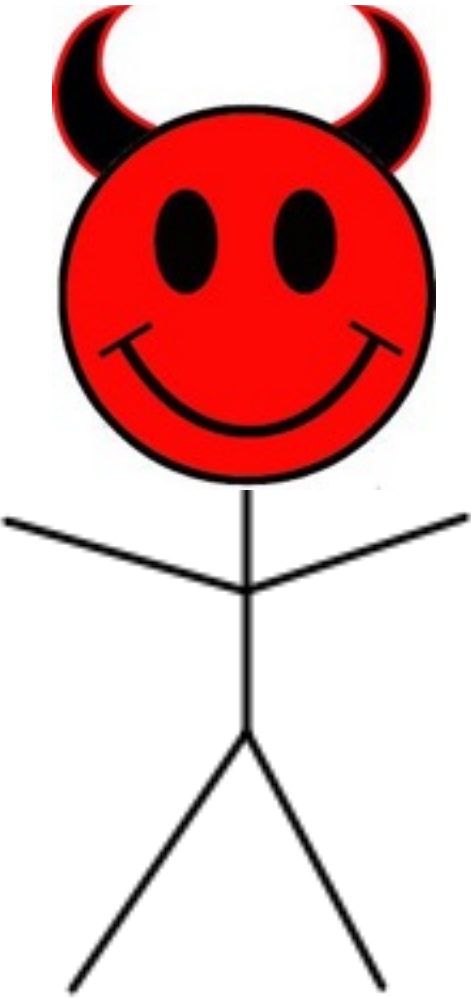
operations is the secret sauce

just not 'traditional operations'

# Traditional IT



dev

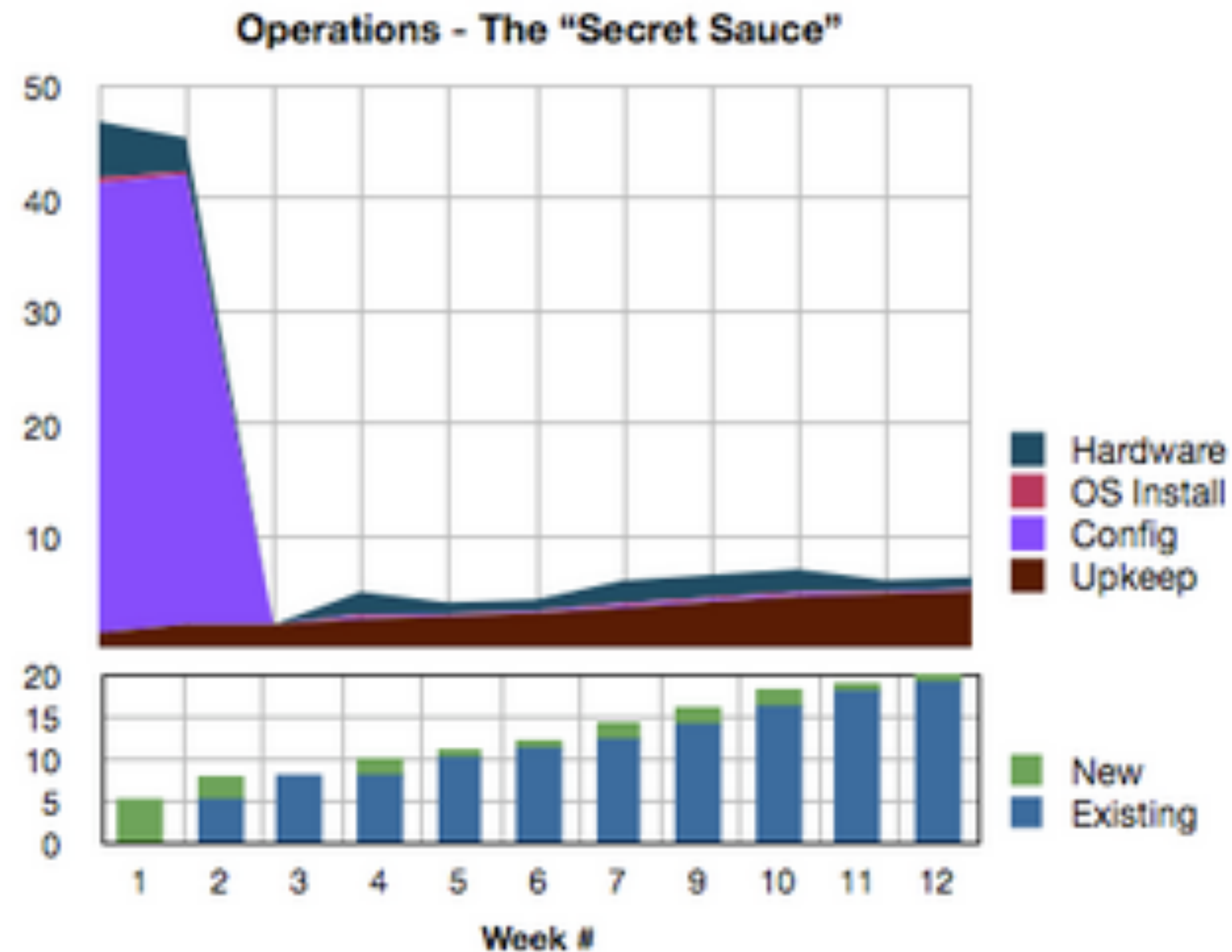
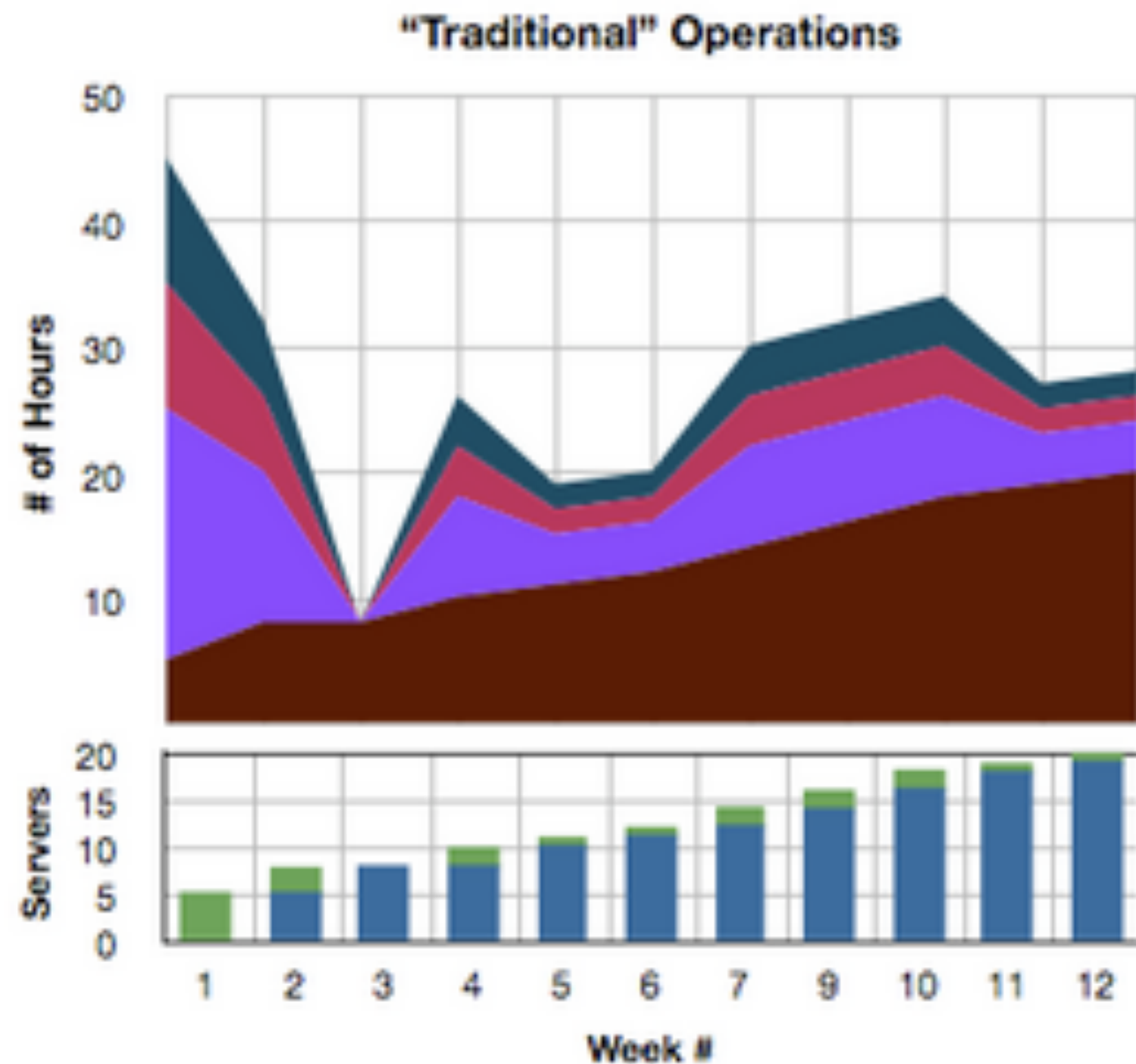


ops

wall of confusion

# Opposing Forces





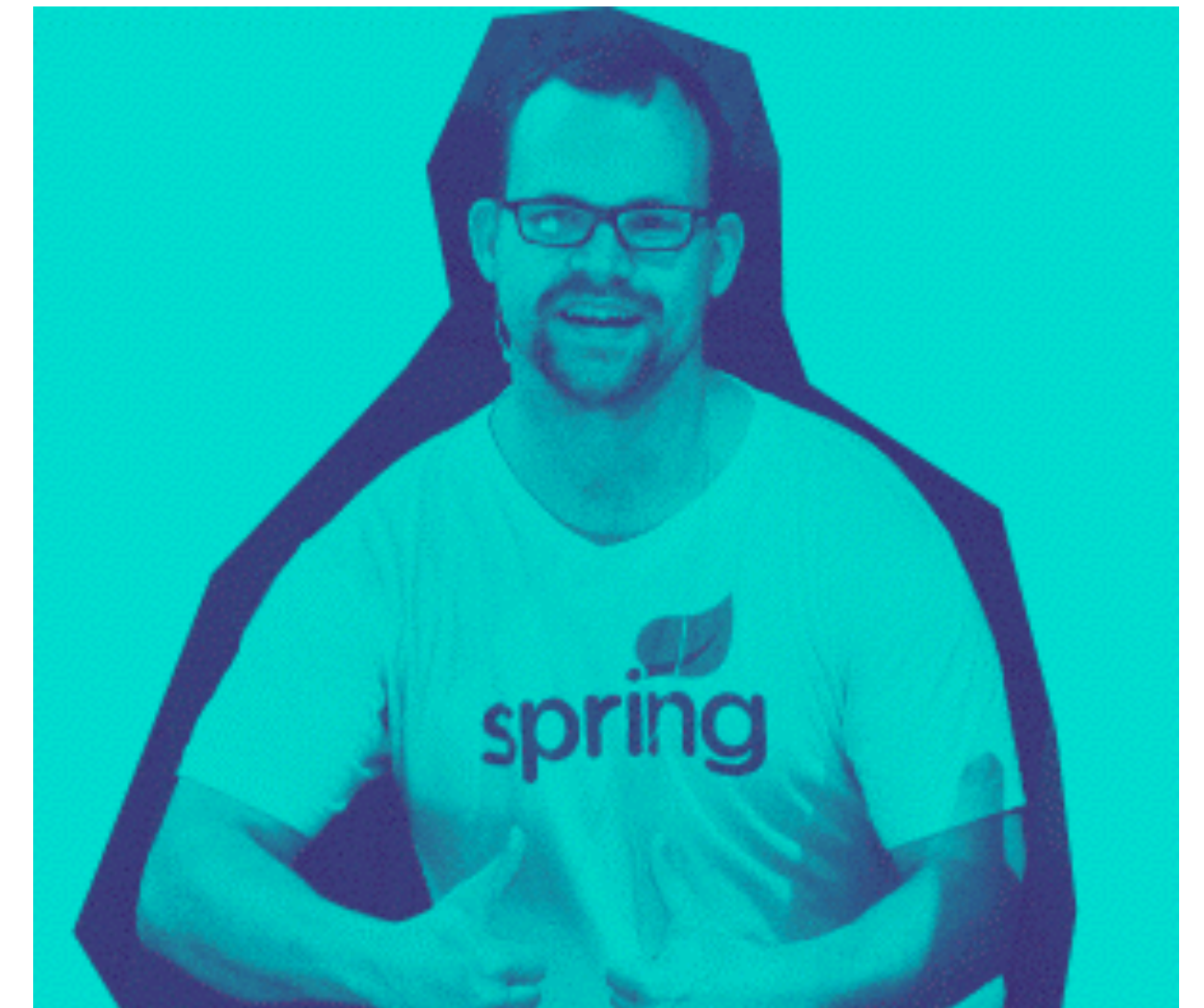
(<http://radar.oreilly.com/archives/2007/10/operations-advantage.html>)

Operations is the secret sauce

“The traditional model is that you take your software to the wall that separates development and operations, and throw it over and then forget about it. Not at Amazon. You build it, you run it.

This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer. This customer feedback loop is essential for improving the quality of the service.”

**–Werner Vogels, CTO Amazon**



if you really love production...

you would carry a pager



# Software's long-term cost

Software engineering as a discipline focuses on designing and building rather than operating and maintaining, despite estimates that 40%<sup>1</sup> to 90%<sup>2</sup> of the total costs are incurred after launch.

<sup>1</sup> Glass, R. (2002). *Facts and Fallacies of Software Engineering*, Addison-Wesley Professional; p. 115.

<sup>2</sup> Dehaghani, S. M. H., & Hajrahimi, N. (2013). Which Factors Affect Software Projects Maintenance Cost More? *Acta Informatica Medica*, 21(1), 63–66. <http://doi.org/10.5455/AIM.2012.21.63-66>



# devops

- developers and operations can and should work together
- system administration evolving to look more like software development
- evolving together as global community sharing solutions

me - in 2010

# devops - calms

- culture
- automation
- lean
- metrics
- sharing

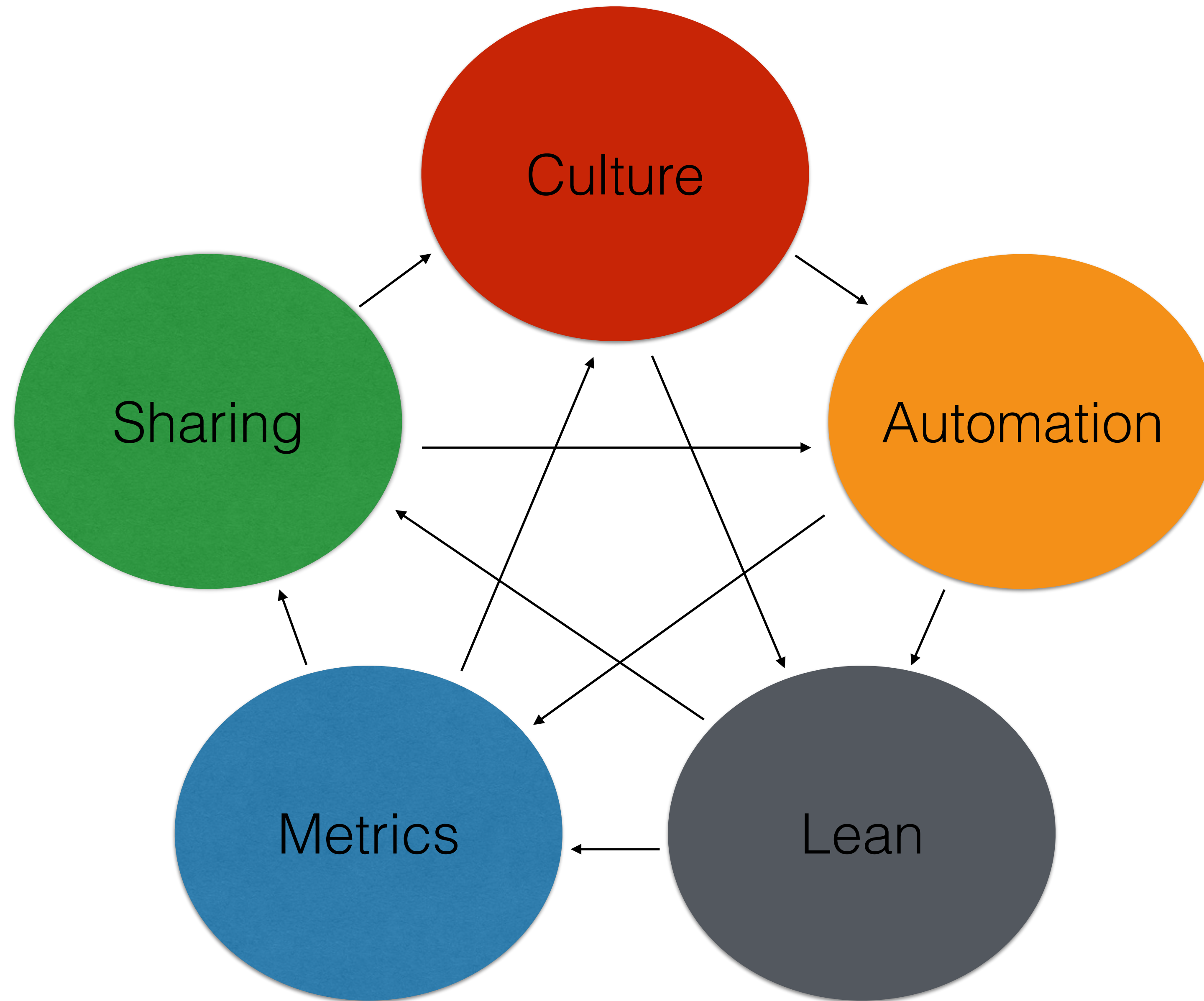


this is the devops

wtf does that mean?

wtf am I supposed to do now?






I'm allegedly writing a book

# Culture



A photograph of Peter Drucker, an elderly man with glasses, wearing a blue suit jacket, white shirt, and dark tie. He is sitting with his hands clasped in front of him, looking slightly to the right of the camera with a thoughtful expression. The background is a plain, light-colored wall.

**‘Culture eats  
strategy for  
breakfast’  
- Peter Drucker**

# Westrum Topology Culture

<b>Pathological (power-oriented)</b>	<b>Bureaucratic (rule-oriented)</b>	<b>Generative (performance-oriented)</b>
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to enquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

align incentives and interests

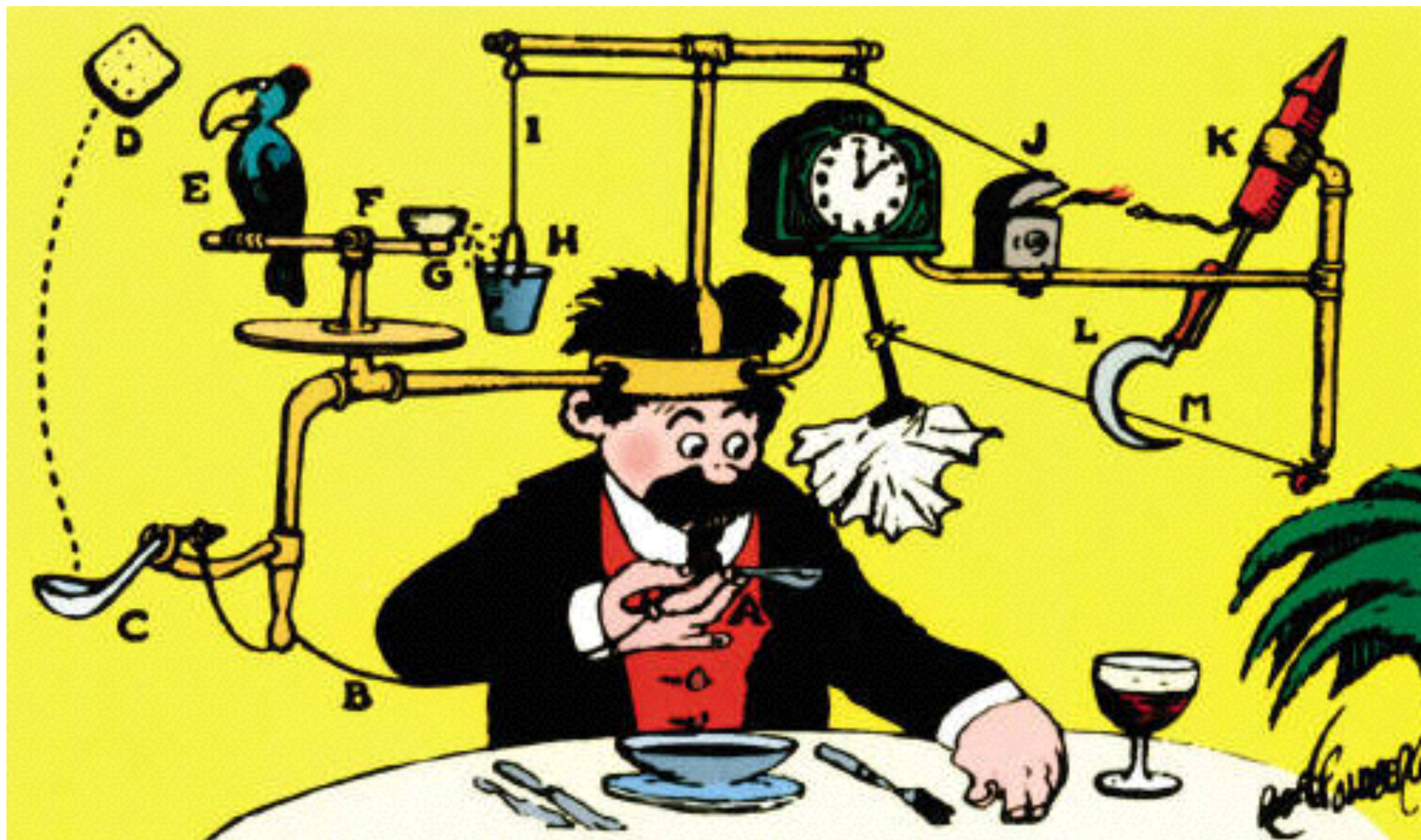
competitive advantage  
vs  
cost center

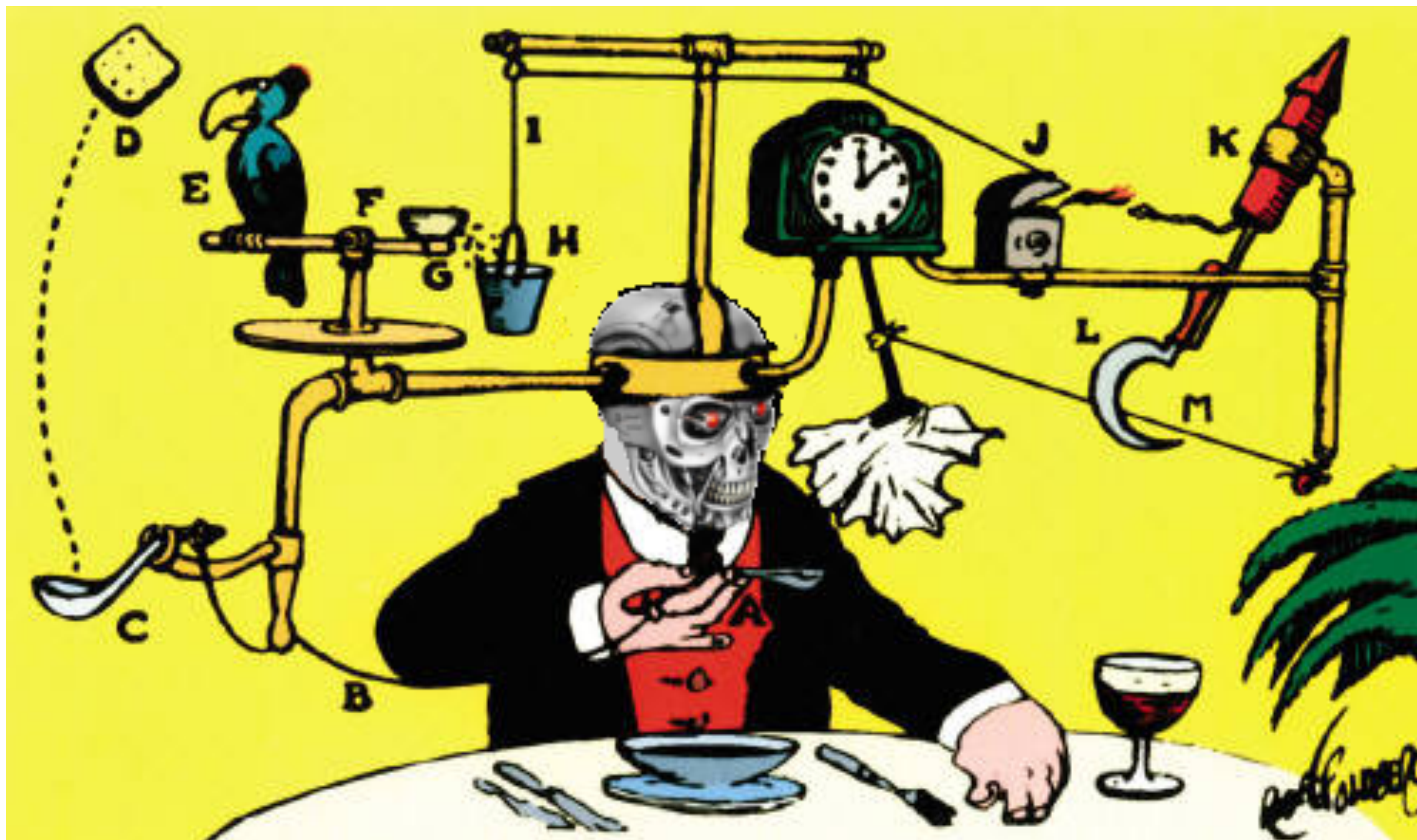
# Automation & Architecture

**AUTOMATE**



memegenerator.net

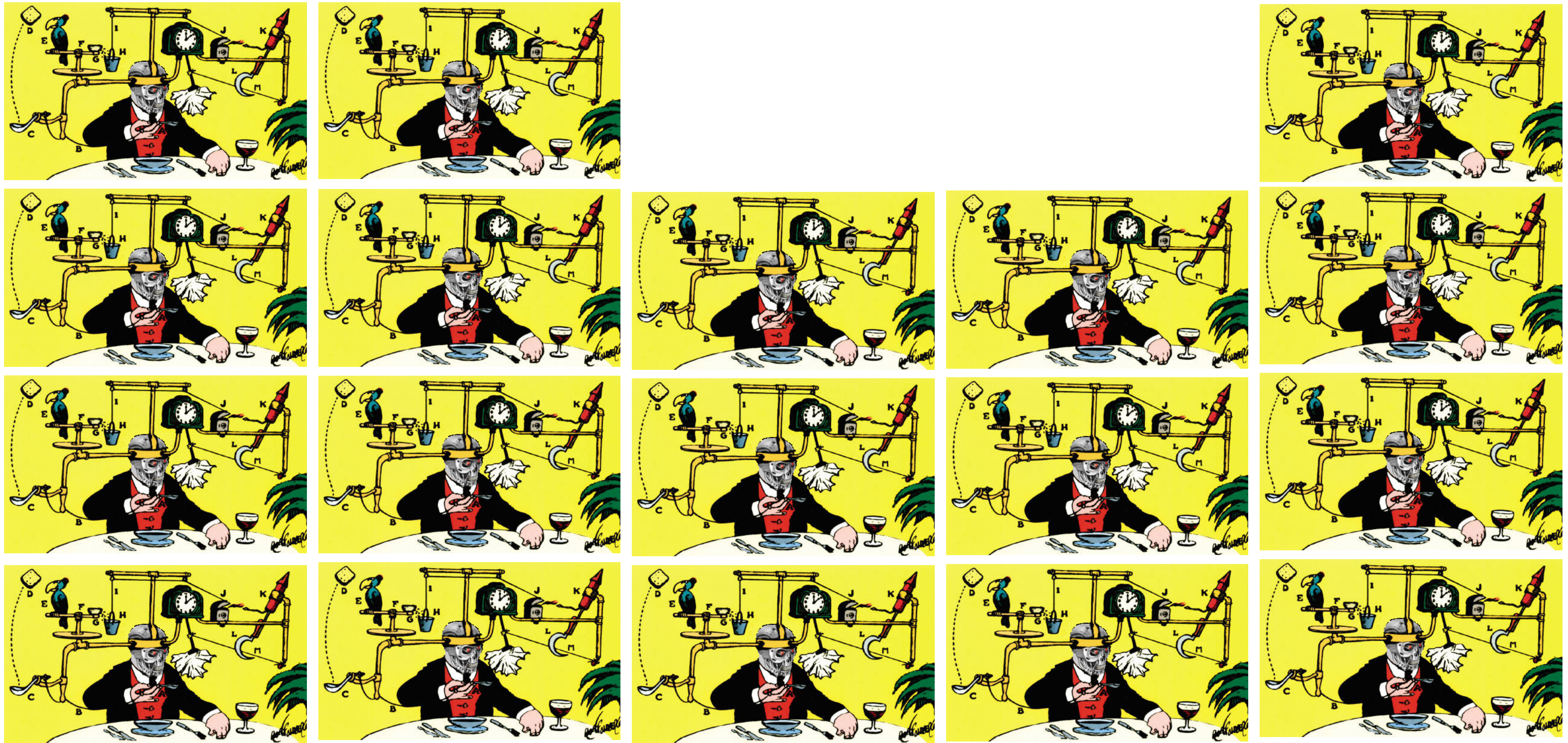




yay automation!



what, how and why you automate  
is as important that you do



yay automation!

If Tetris Has Taught Me Anything,  
It's That Errors Pile Up and  
Accomplishments Disappear

I'm sure some of you  
have lived this too

or are about to

## Manual

## Scripted

## Platform

toil

effort

directed

catastrophic failure

disaster recovery

self healing

incidents

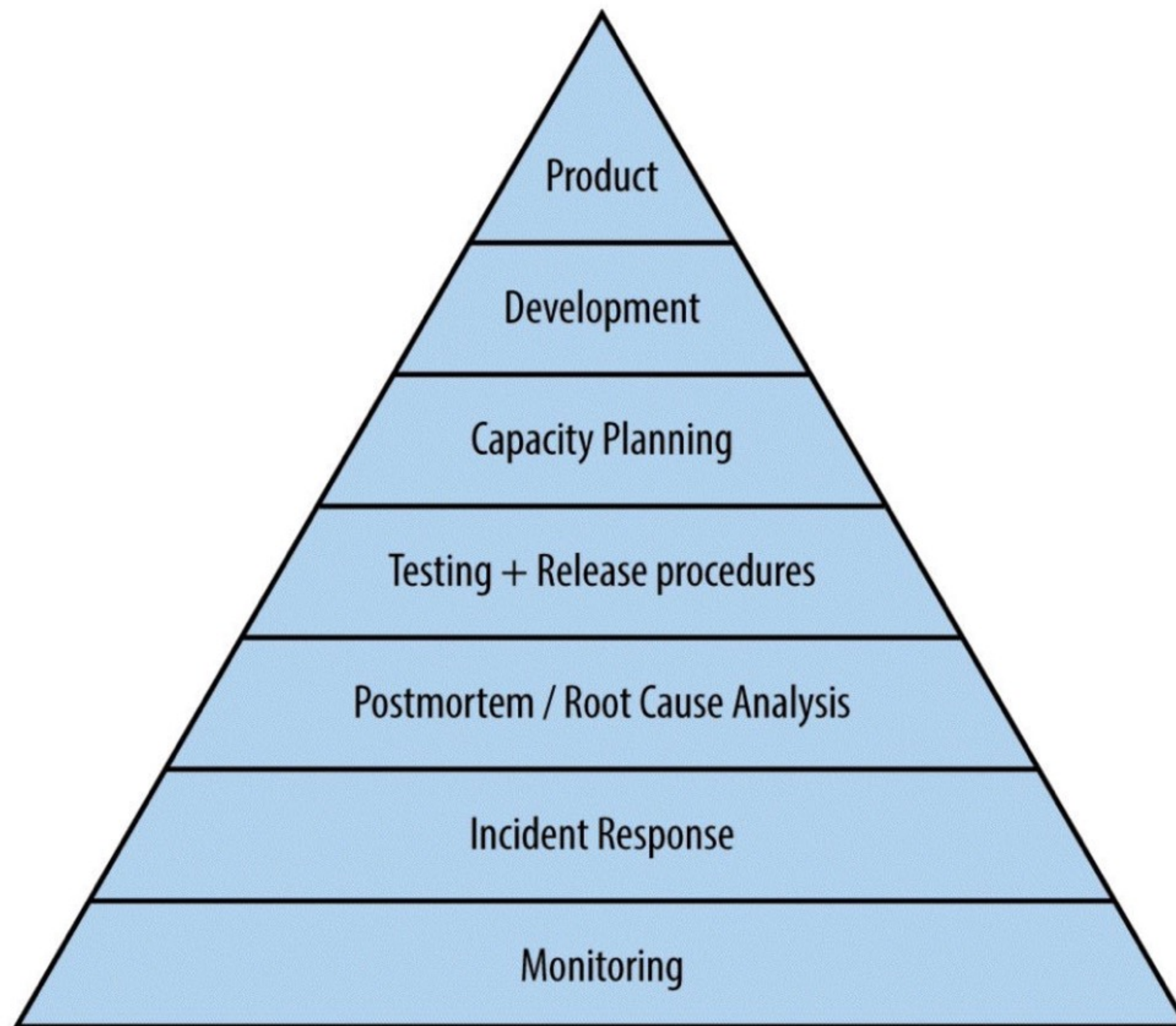
MTTR

continuous partial failure

Almost every task run under Borg contains a built-in HTTP server that publishes information about the health of the task and thousands of performance metrics



# Metrics



*Figure III-1. Service Reliability Hierarchy*



# Service Level Objectives

what are your objectives?

how do you know you are meeting them?

**unmonitored**

**measured**

**insightful**

no info

data

SLI

ssh

aggregation

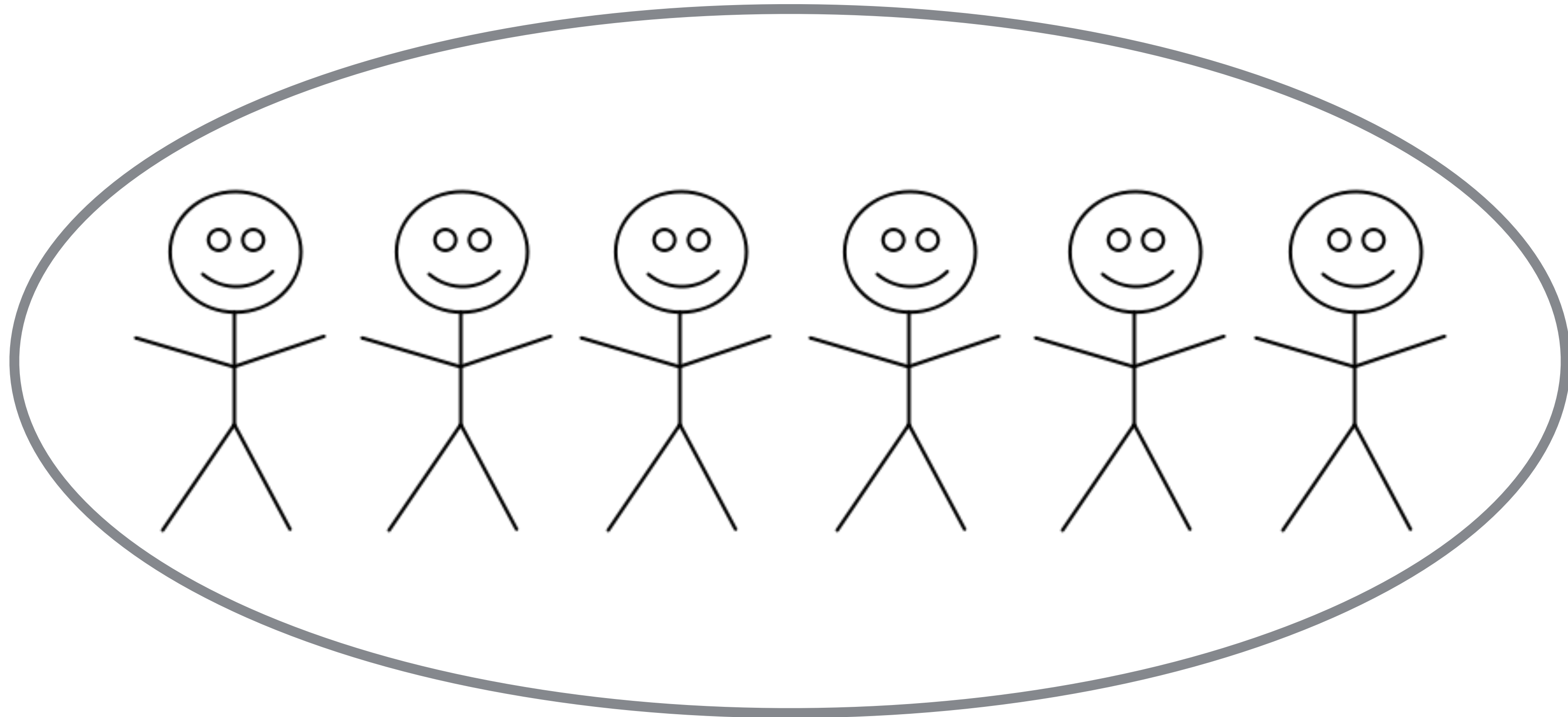
dashboards

never gets done

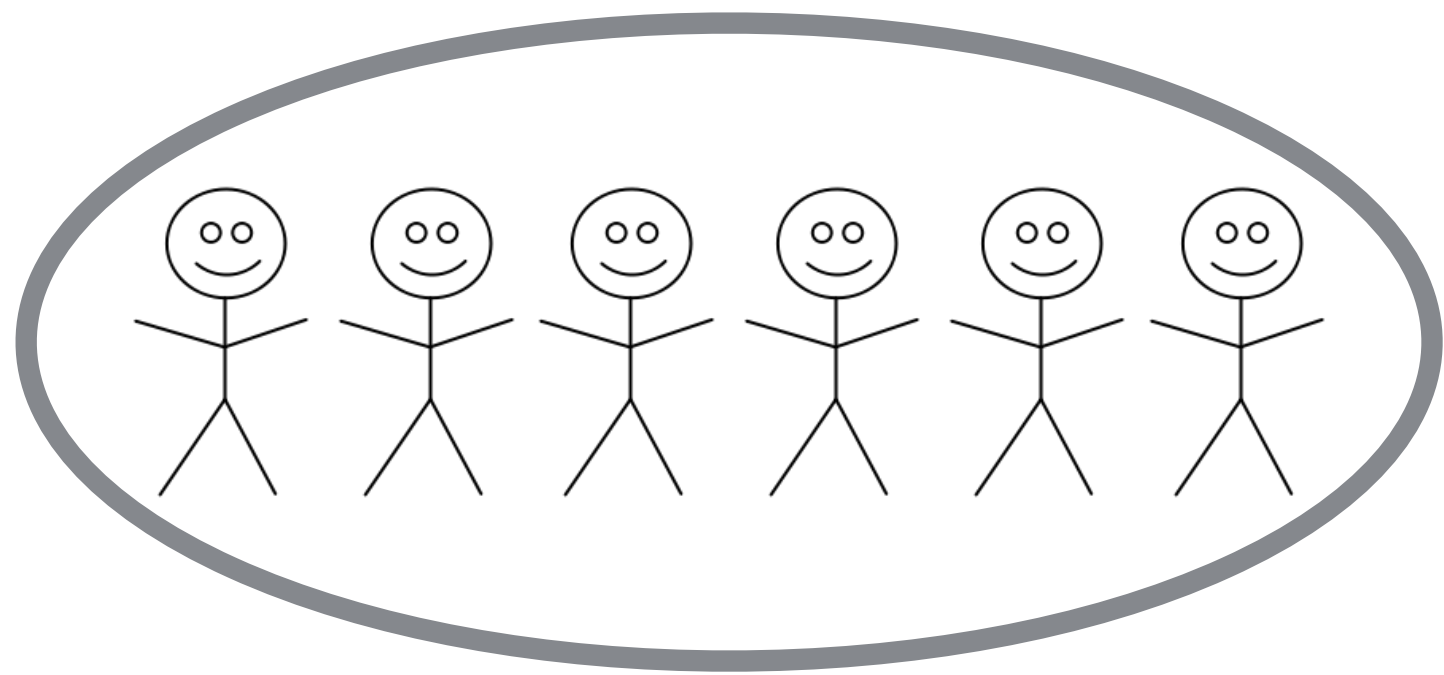
secondary

built in

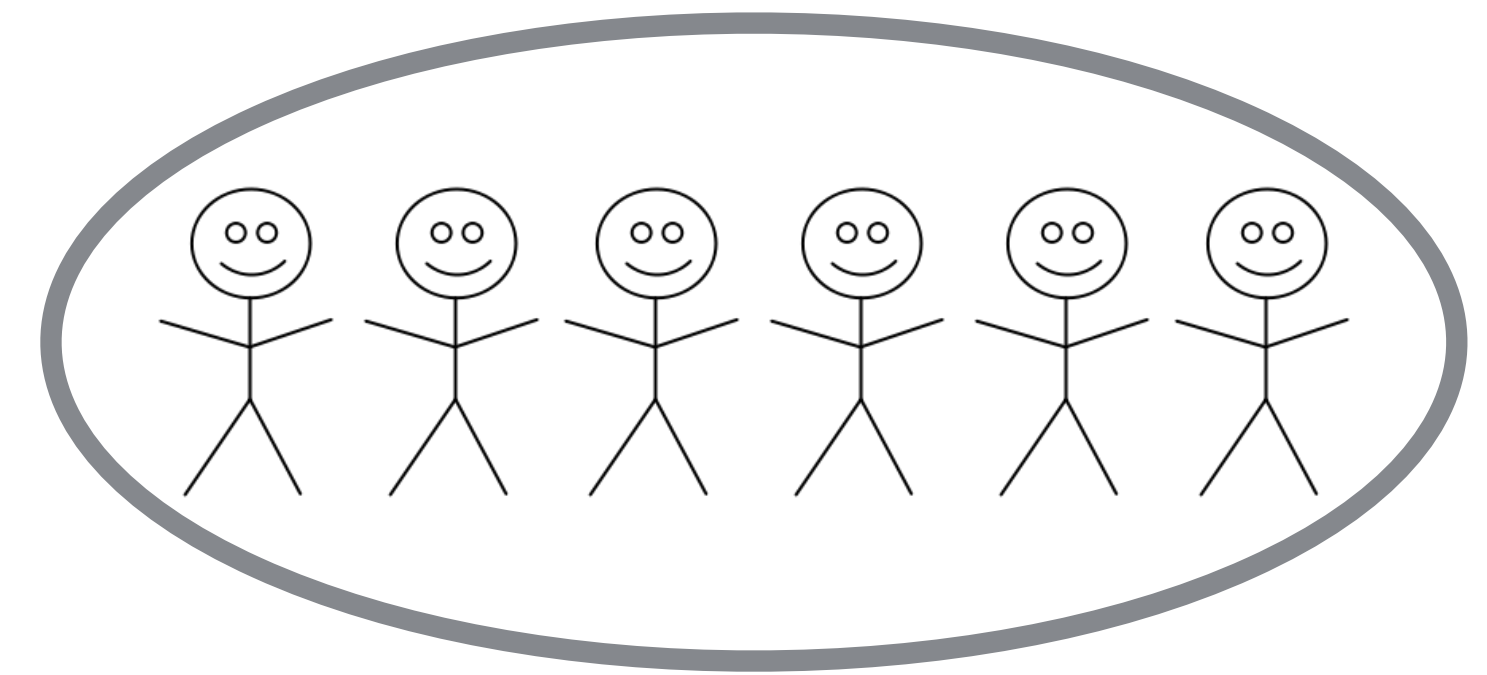
# Sharing



Global Community of Practice

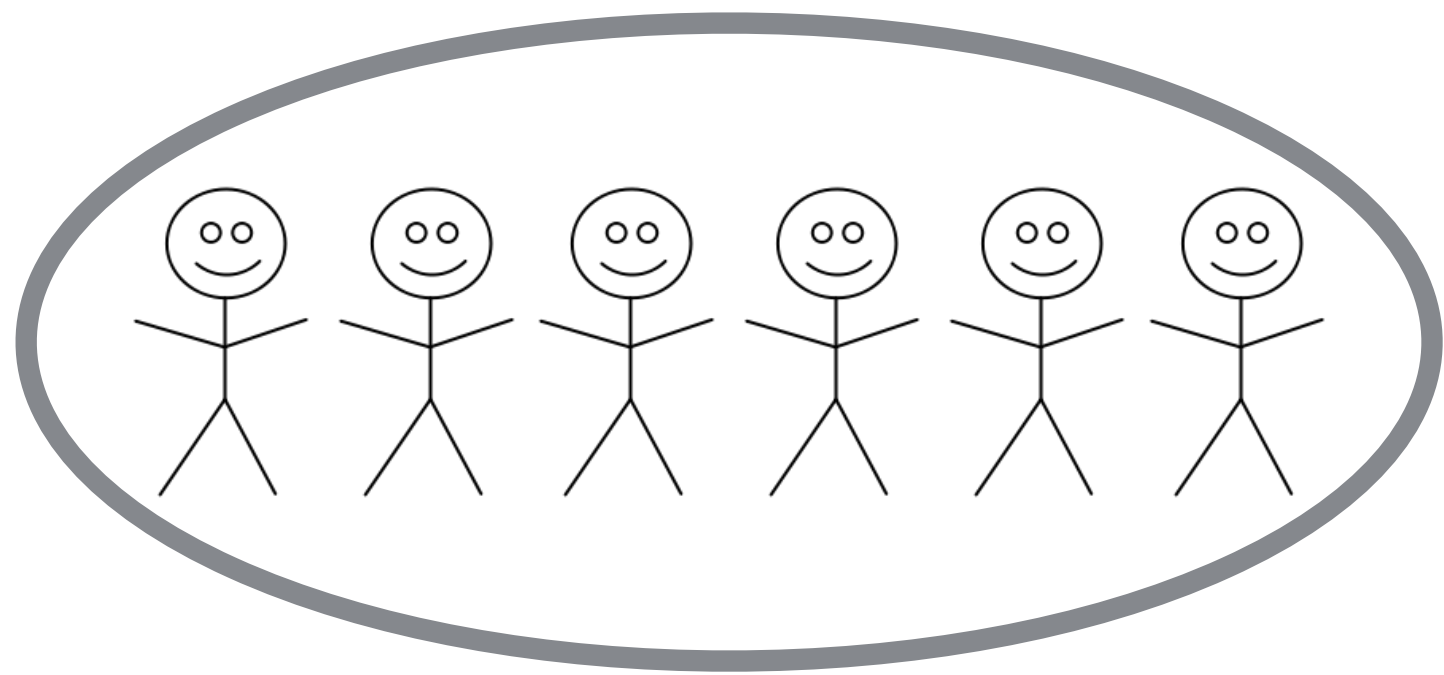


dev

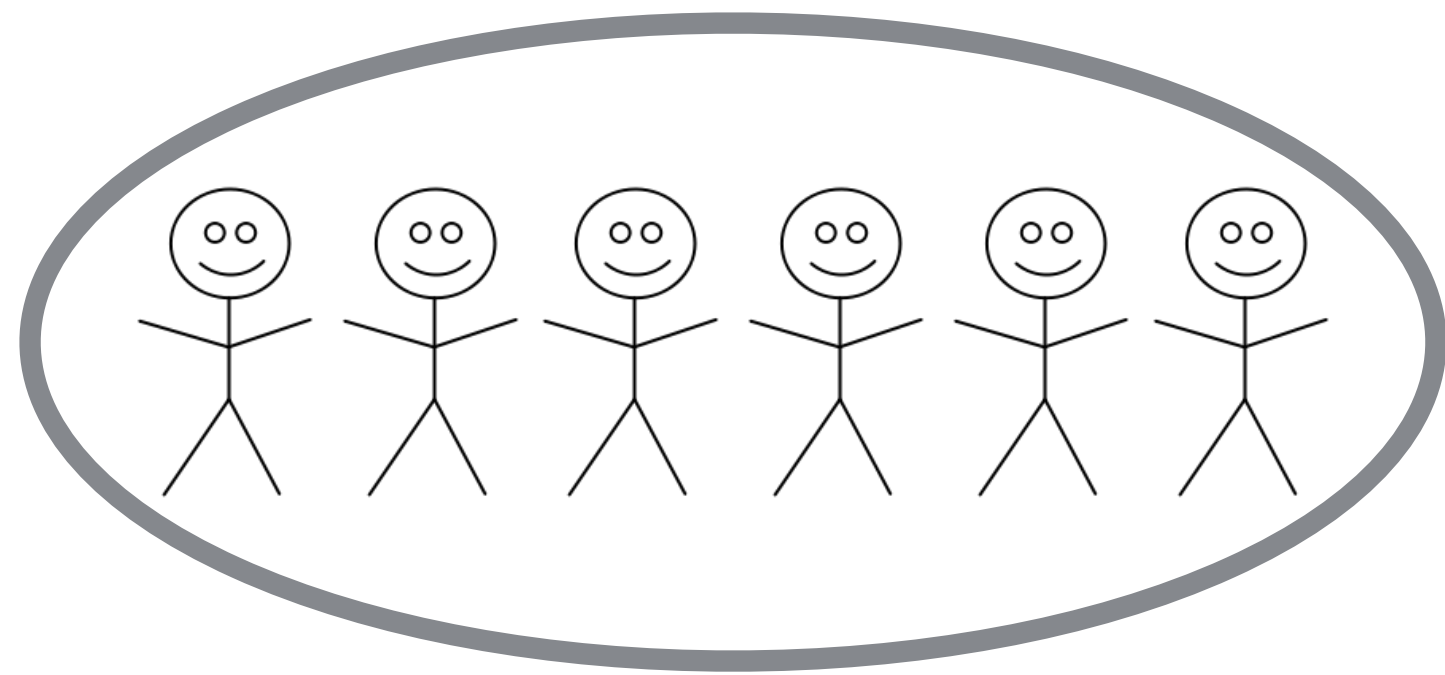


ops

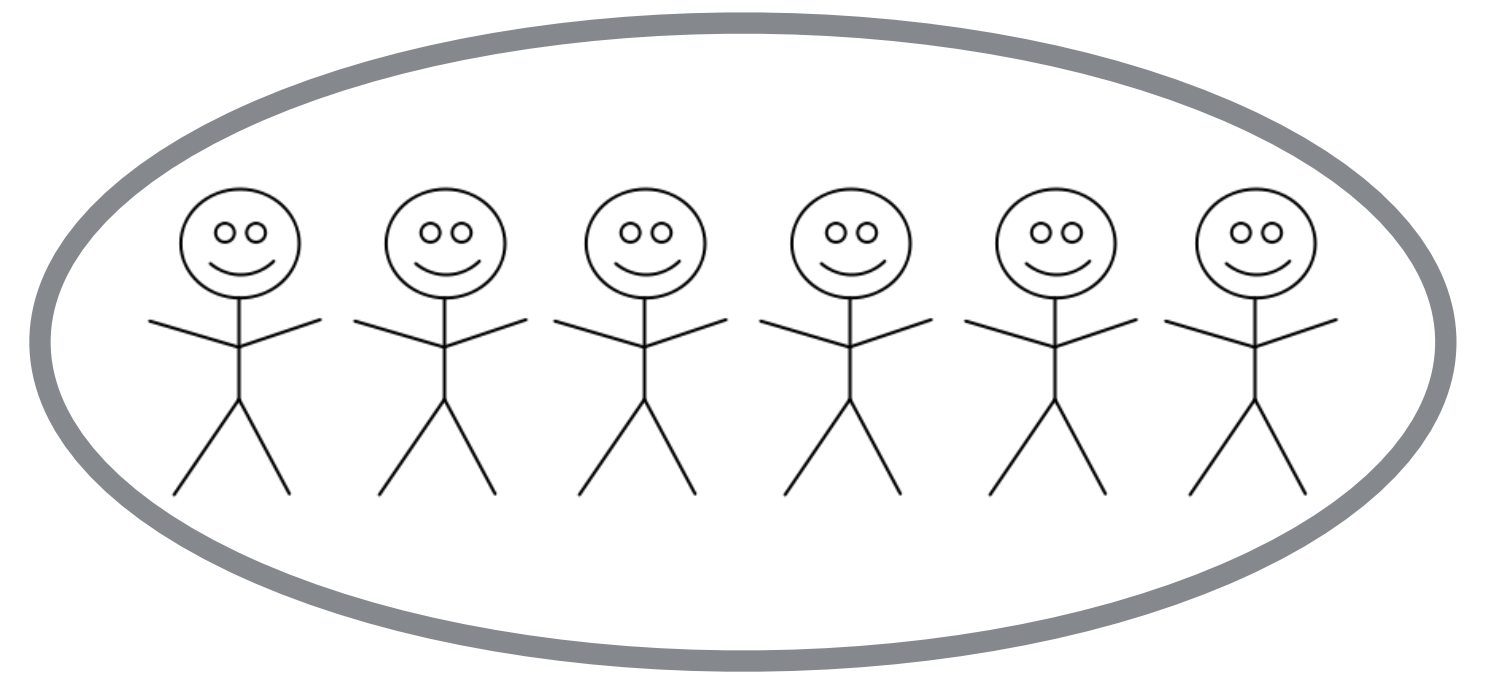
wall of confusion



dev

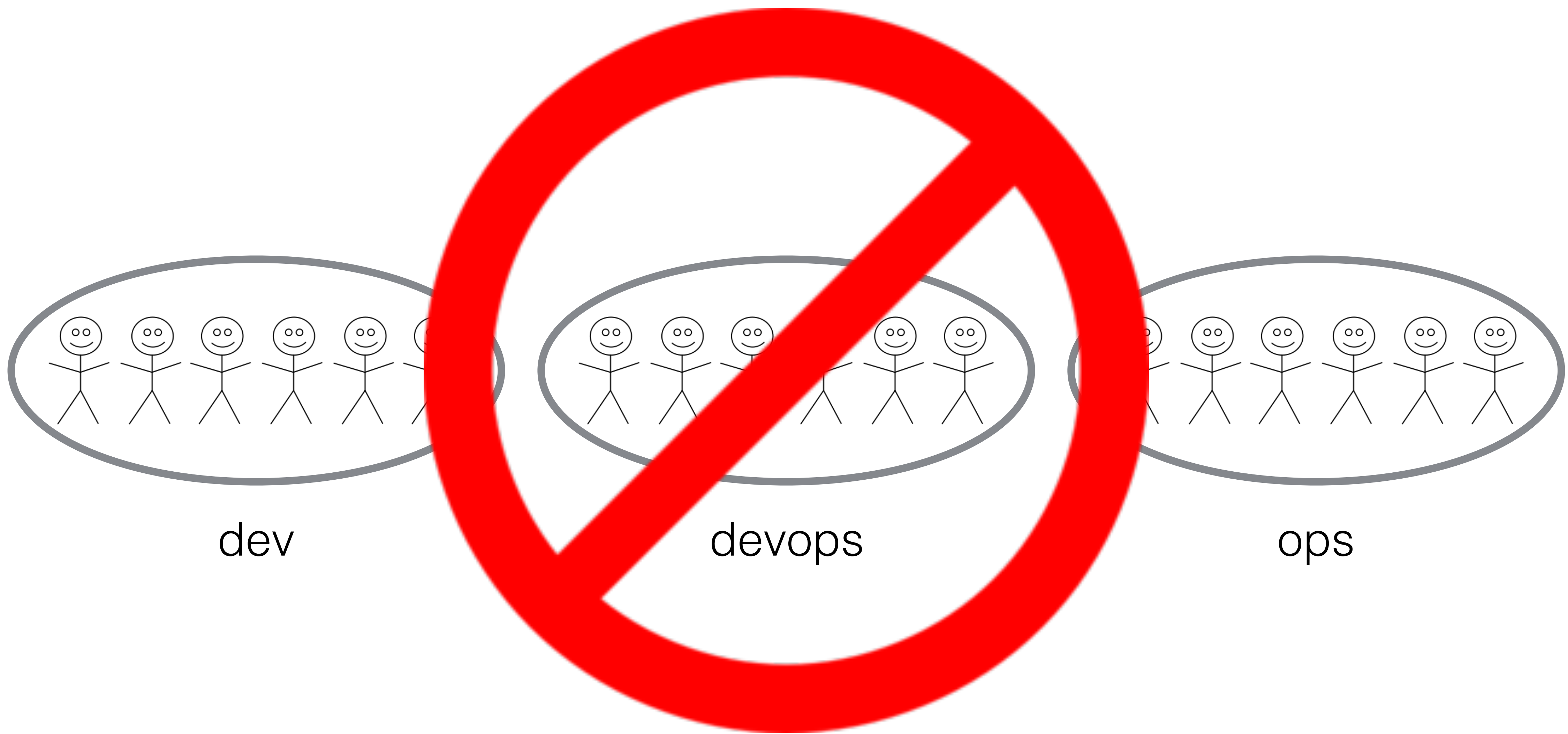


devops



ops

YAY!!



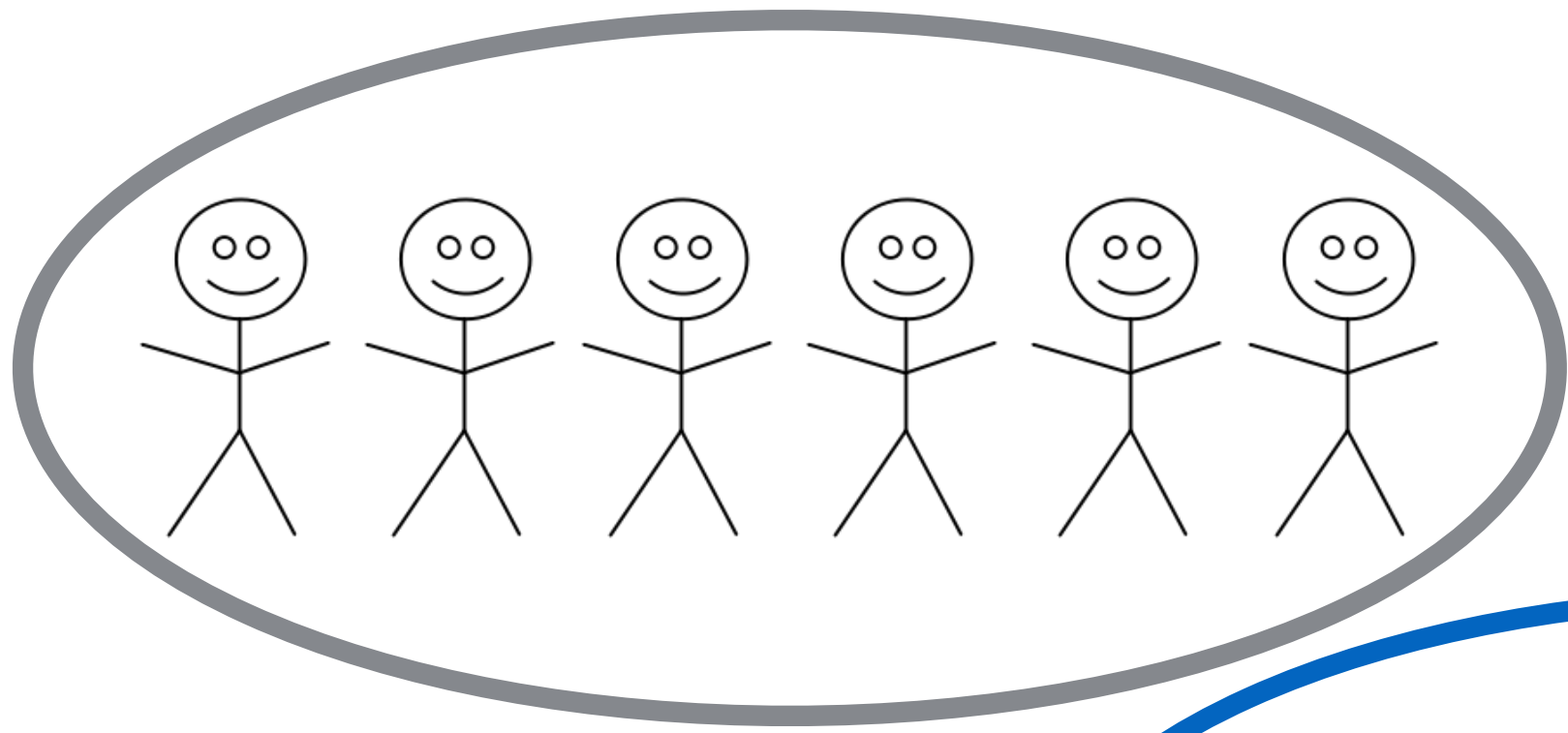
dev

devops

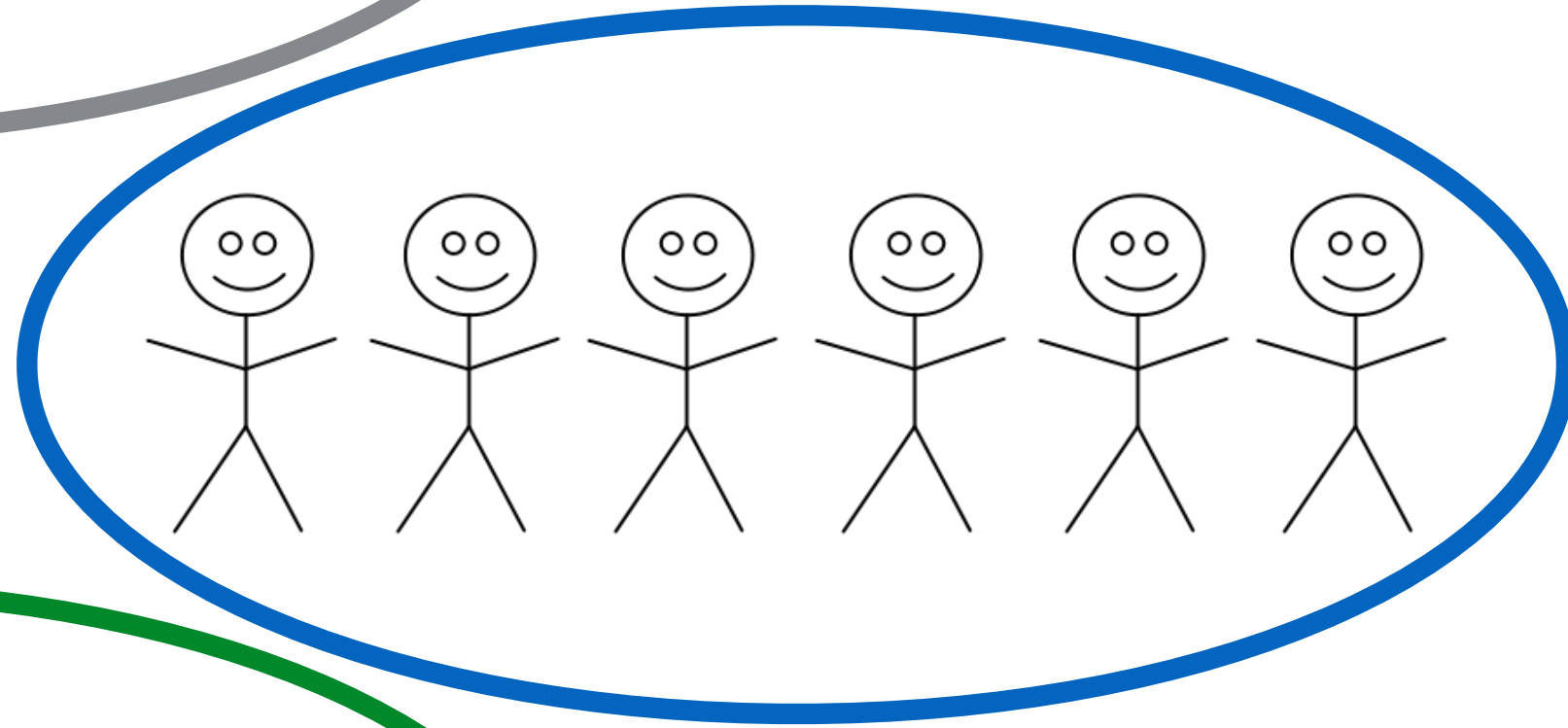
ops



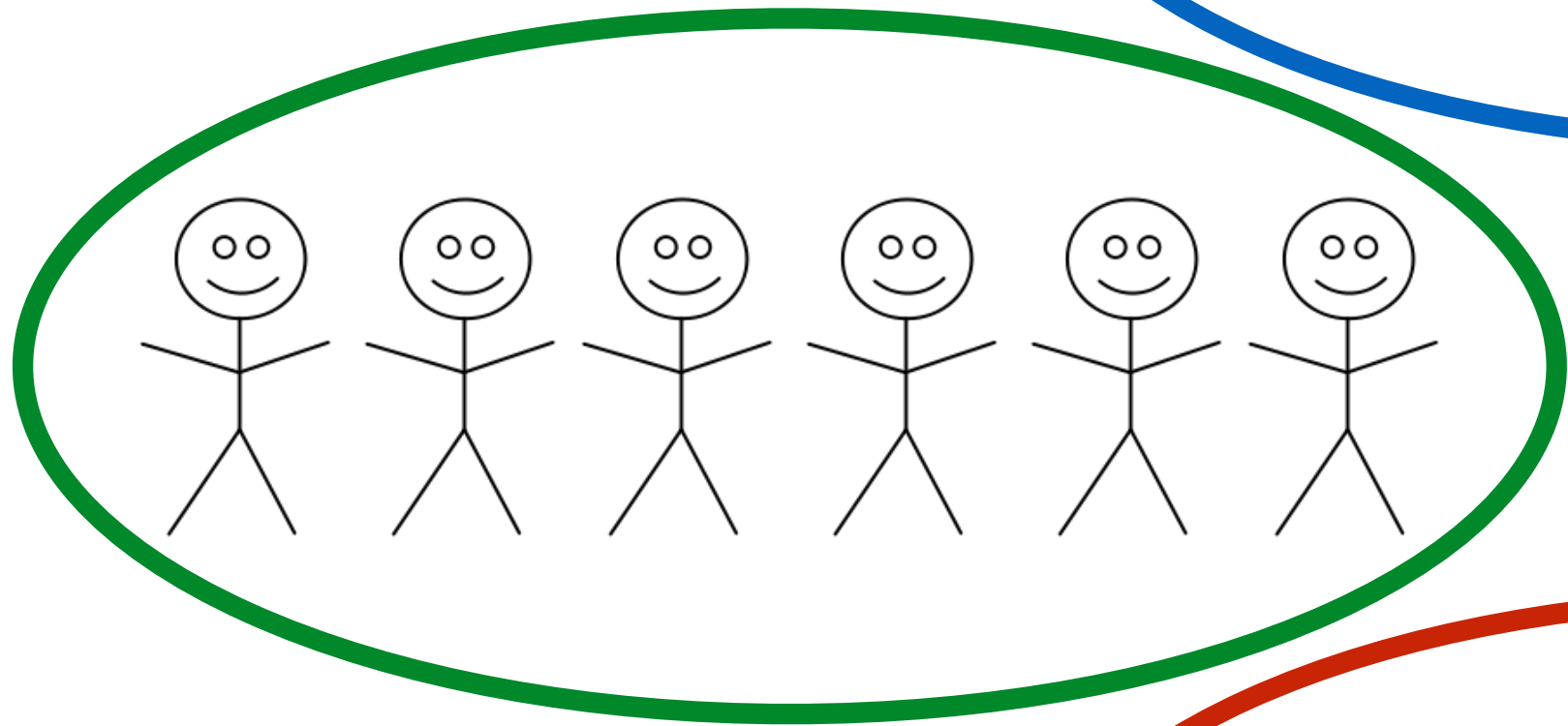




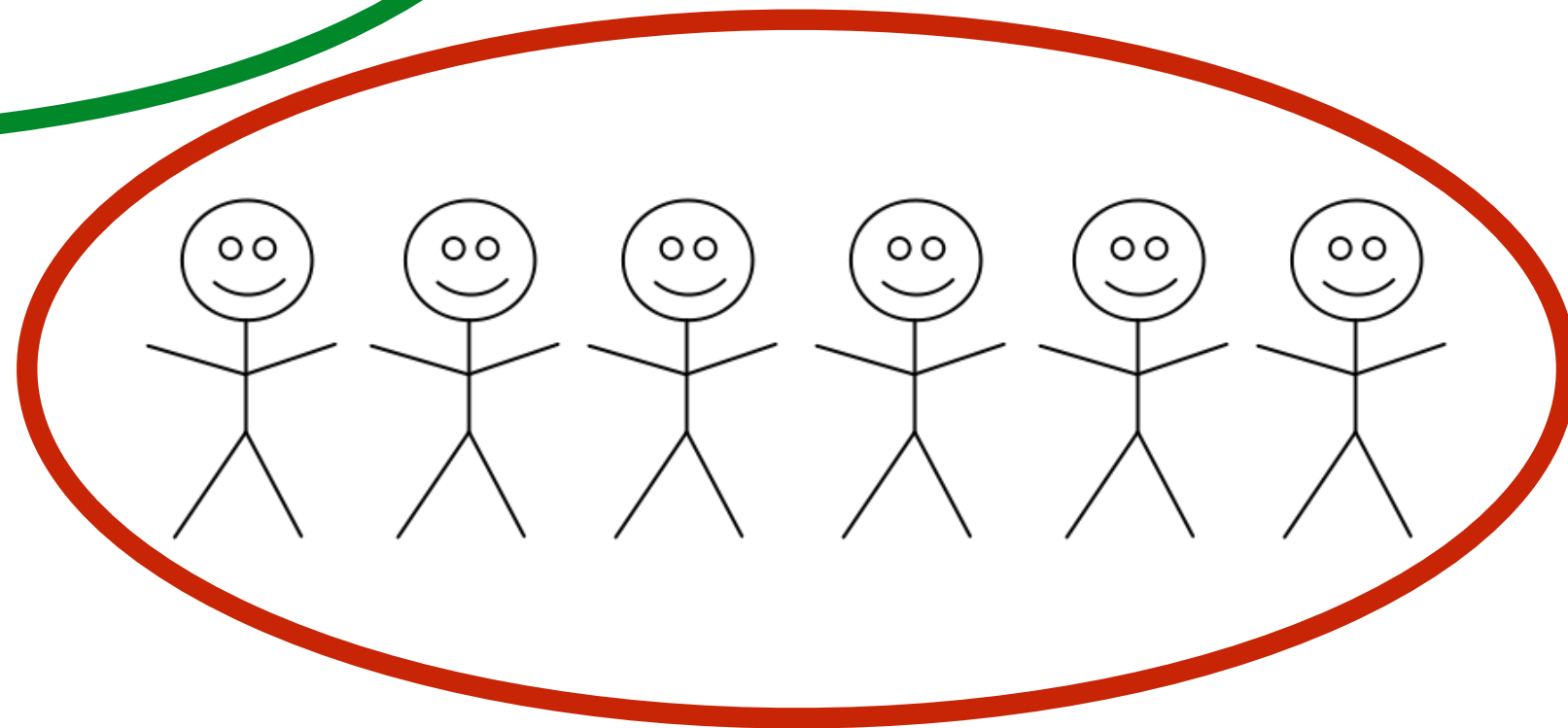
developers



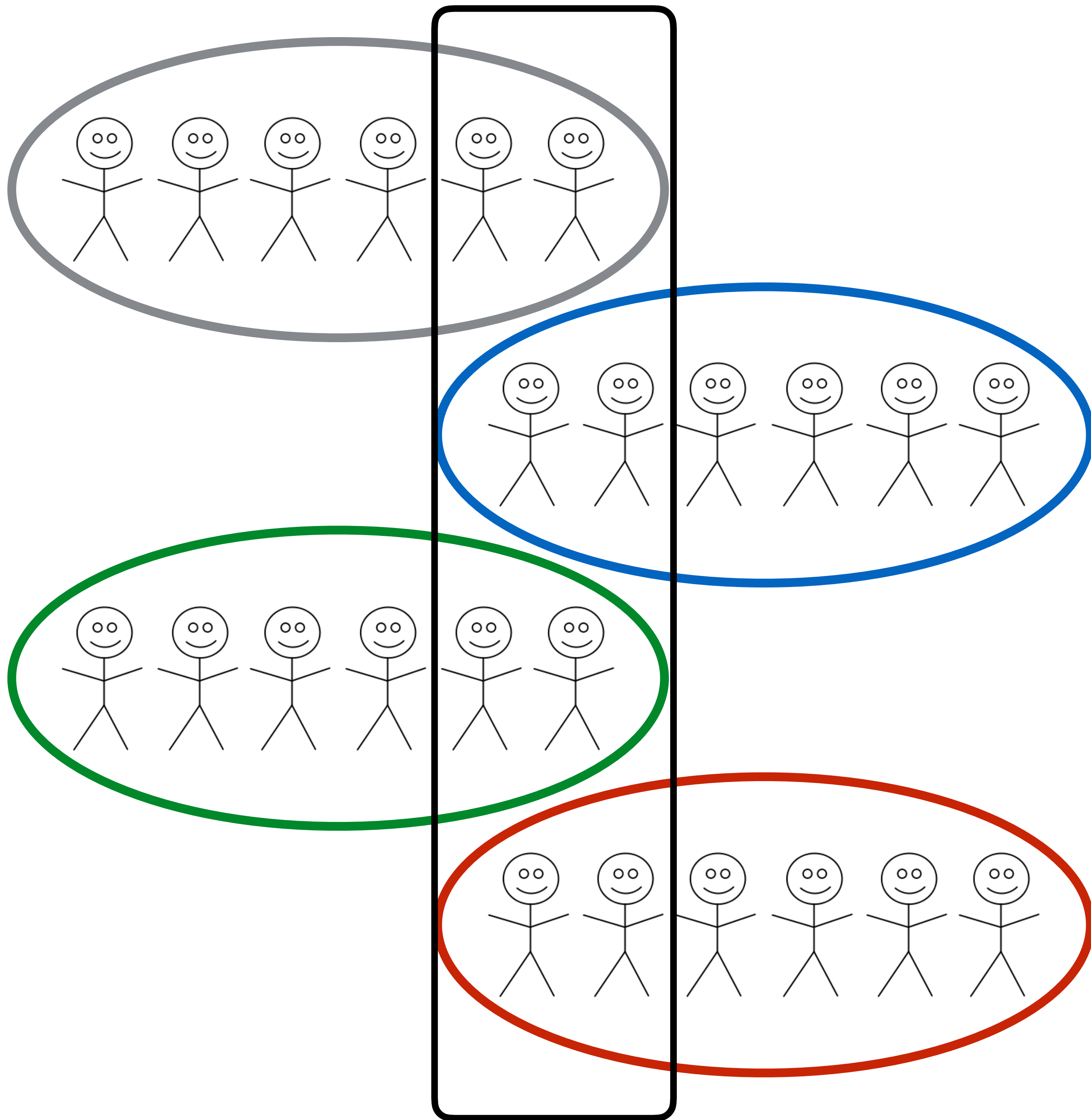
operations



business



security



community of interest

**hidden**

**available**

**ambient**

can't find

searchable

cultivated

strong silos

publish info

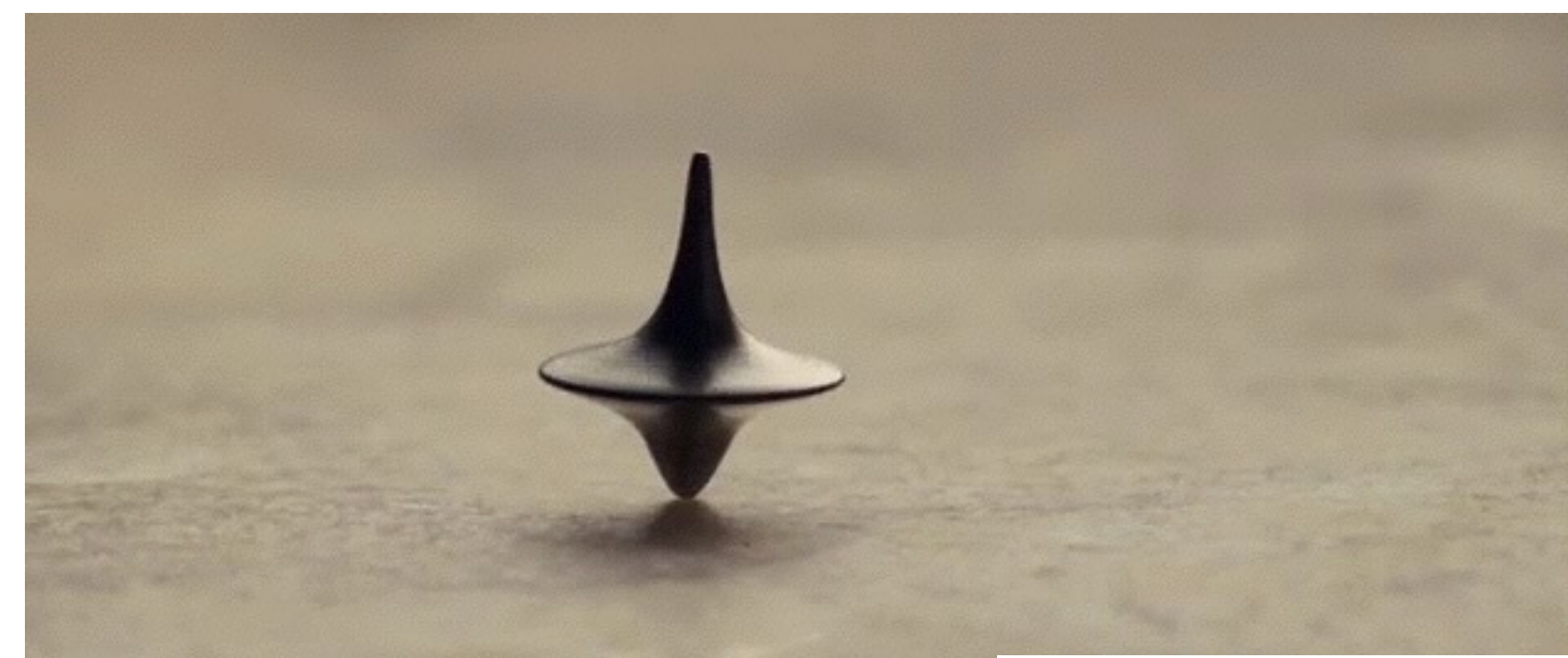
share personally

everything is secret

secret to company

global community

# Lean



Lean Subsumes ALL the Things

# Continuous Improvement

改善

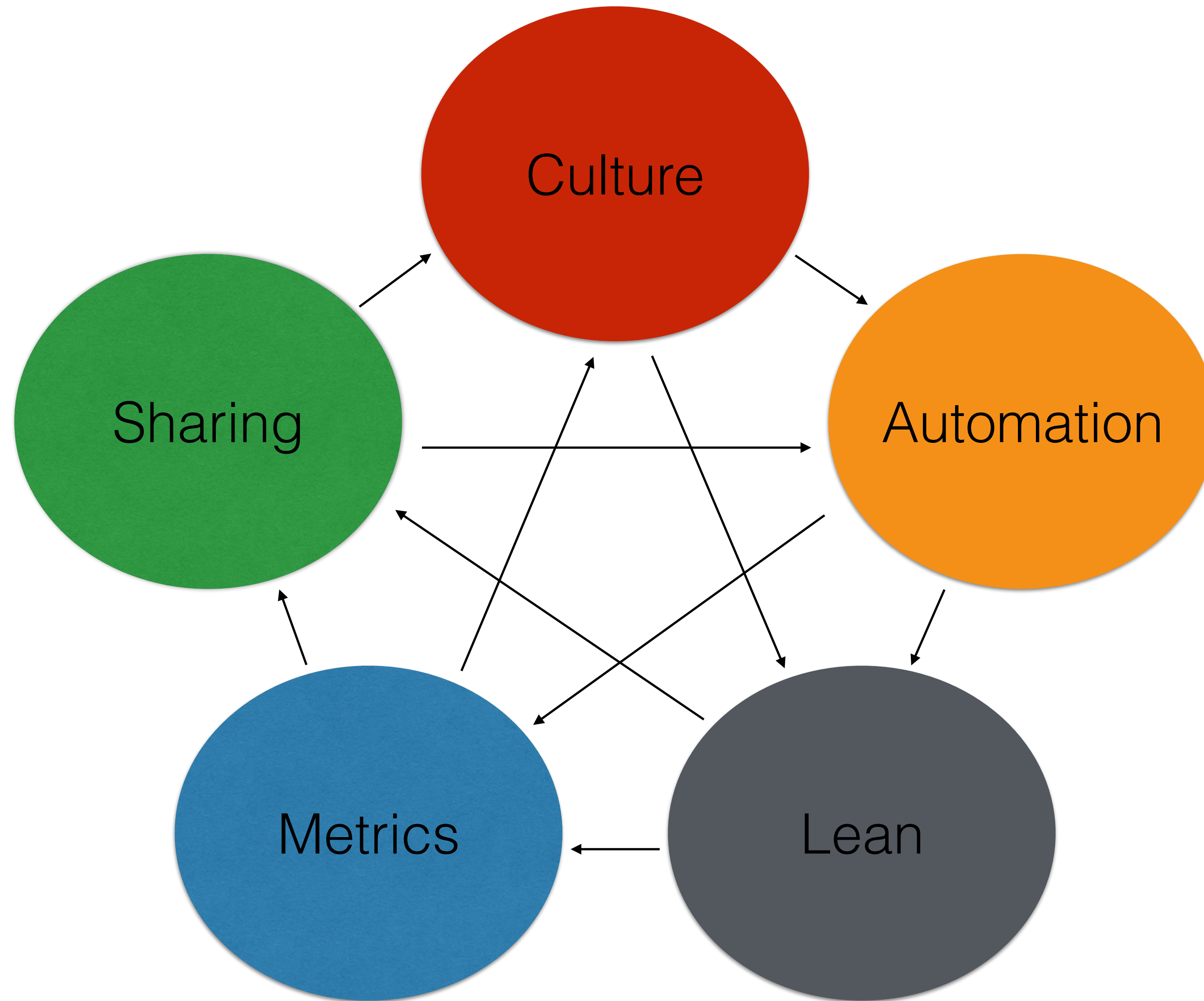
Kai = Change

Zen = Good

CALMS sounds better than CAMS

「(ツ)」





**culture**

**automation**

**lean**

**metrics**

**sharing**

pathological

manual

isolated

nothing

hidden

bureaucratic

discrete

systemic

measurement

available

generative

continuous

exceptional

insight

ambient

# mixing the elements

- pathological cultures can be automated
- automation can be unmonitored
- metrics can be hidden
- generative cultures can toil
- good enough can stagnate



everyone wants the devops

Well actually...

# what they really want

- scalability
- availability
- reliability
- operability
- usability
- observability
- all for free
- without changing anything

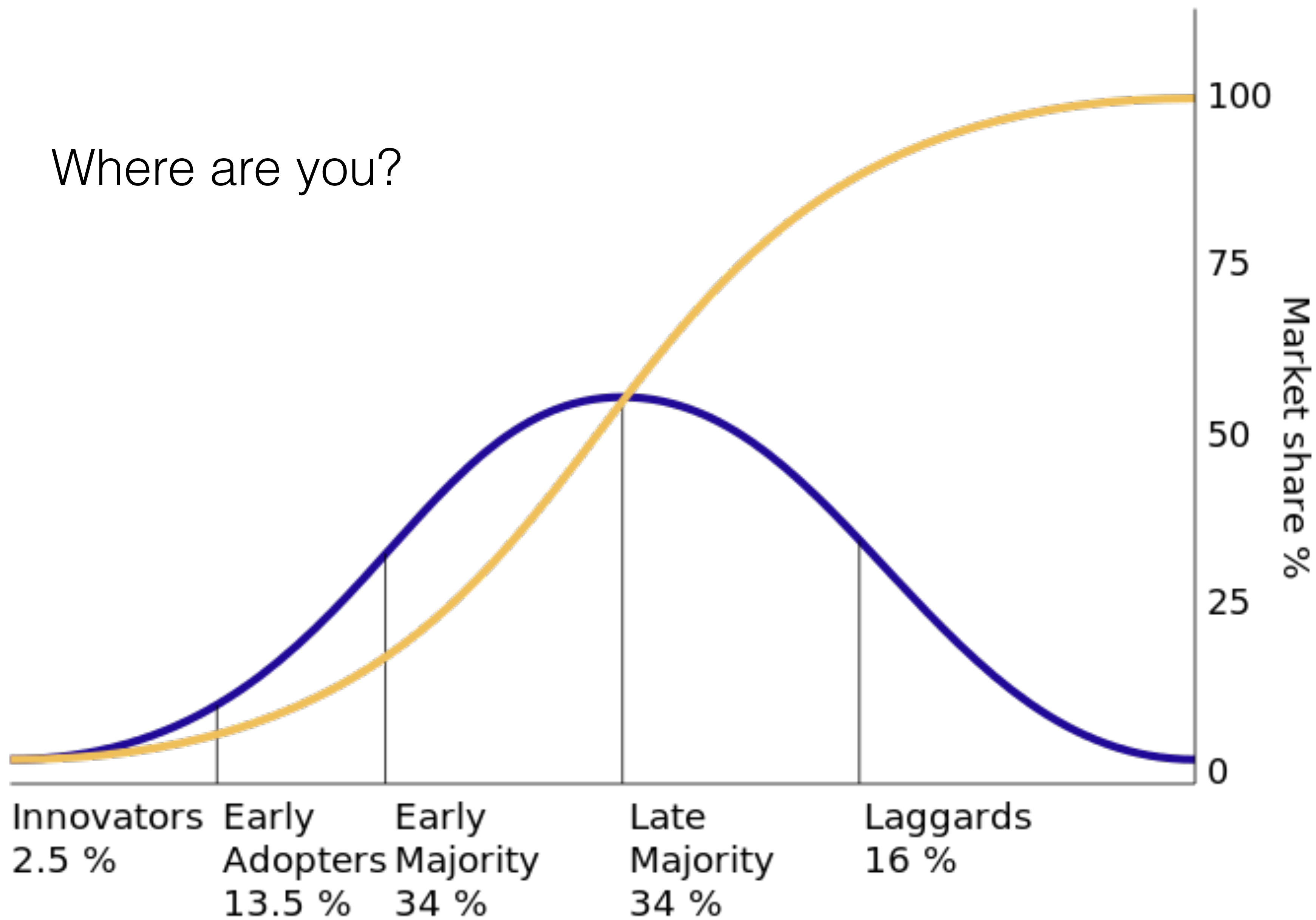
without changing anything

without changing anything

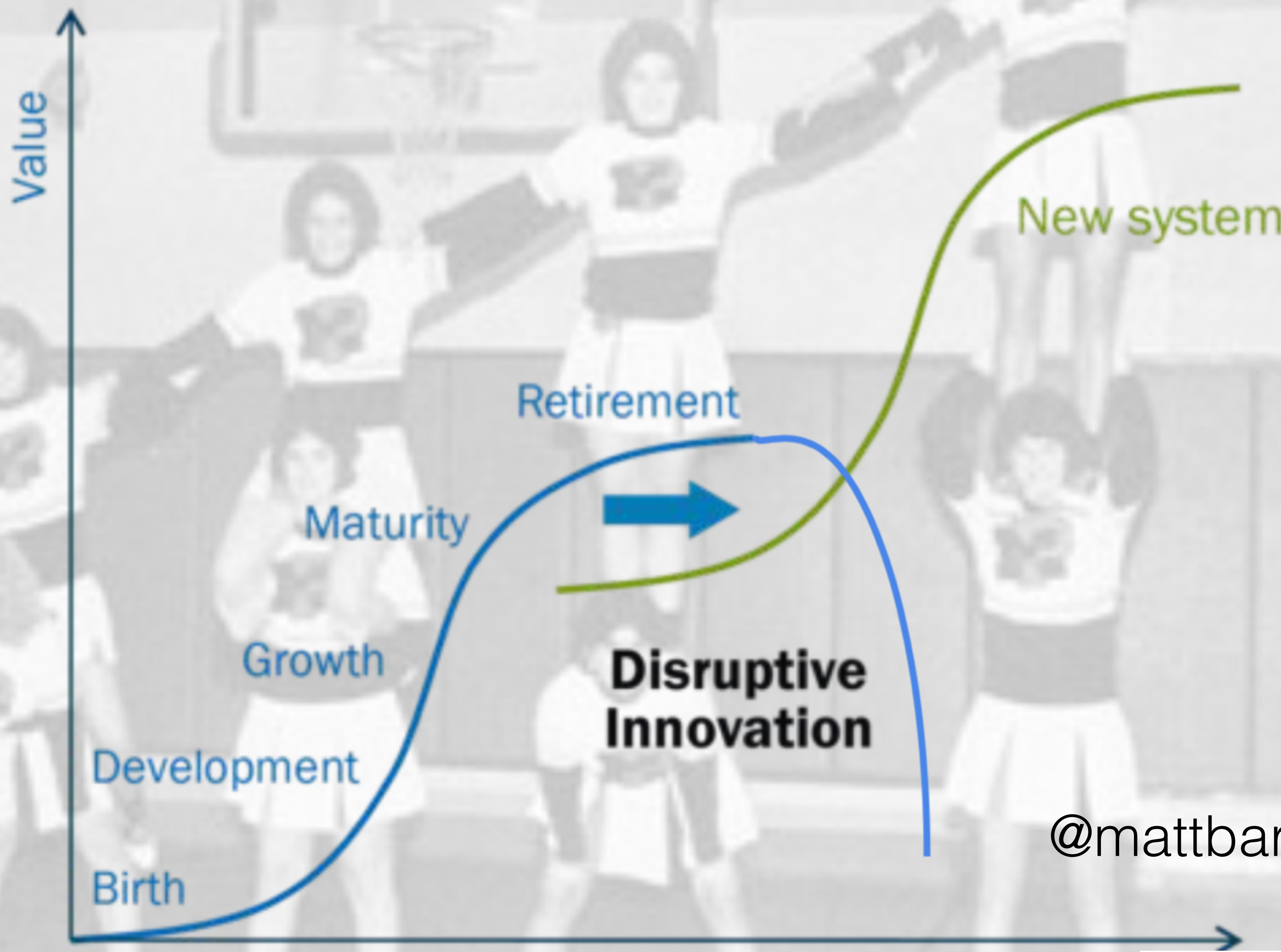


without  
changing anything

Where are you?



# Innovations Build on Each Other



@mattbarcomb





富嶽三十六景 神奈川 浪裏

六舟 江戶 一



WARNING:  
surfing is hard

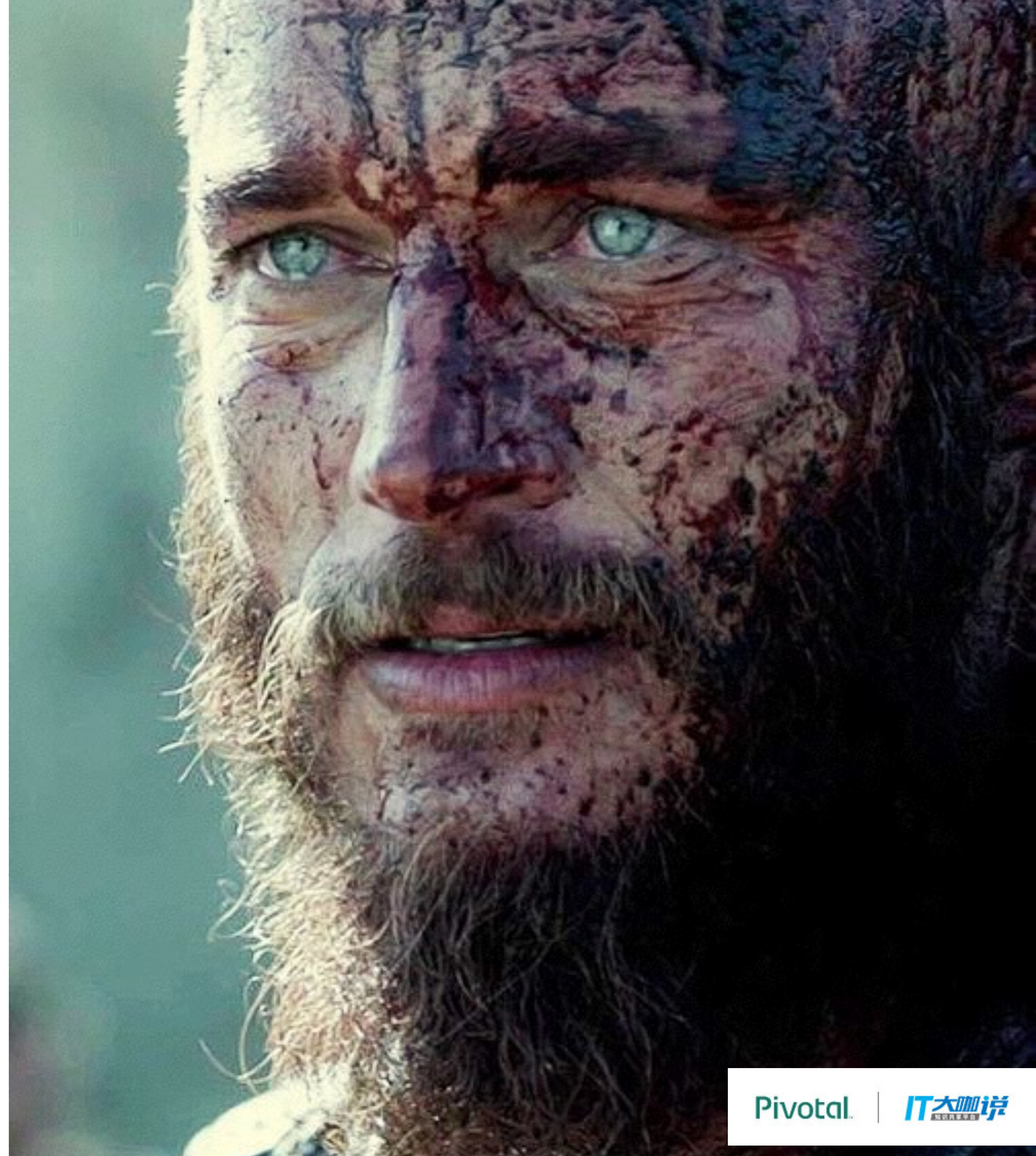
WARNING:  
software is hard



software was always hard

often feels like this

---





# Man Shouts “F\*ck That Alligator”, Jumps Into Lake And Is Killed By Alligator



# the 5 stages of devops

- Denial
- Anger
- Bargaining
- Depression
- Acceptance



optimizing human performance and  
experience operating software...

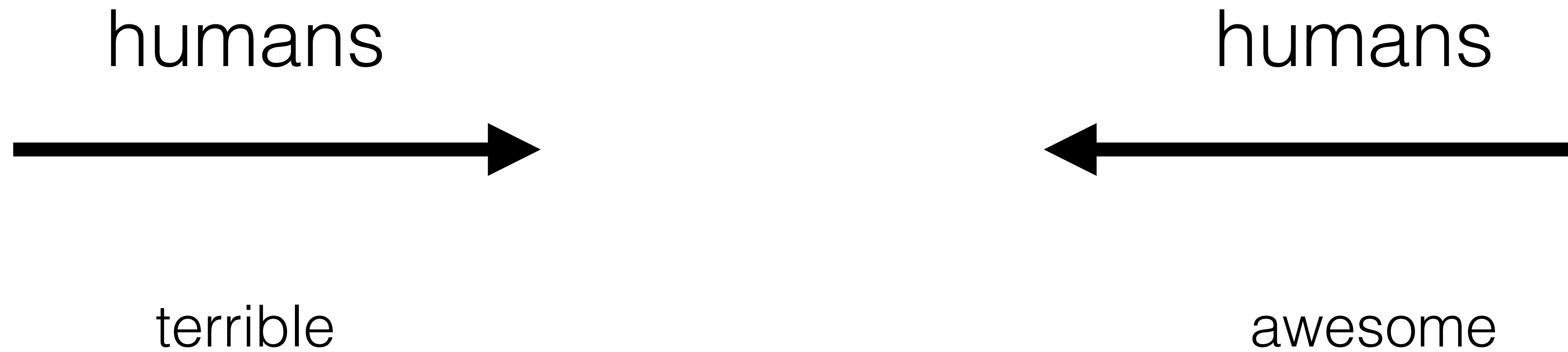
with software...

and humans

software is hard

humans is harder

# Opposing Forces





you are probably a human

humans can learn

You haven't learned anything  
until you change your behavior

“I don't have time to learn new things because I'm too busy getting things done!”

- least productive person in the world

saying 'devops' doesn't fix  
pathological culture

saying 'devops' doesn't fill  
a lack of vision

saying 'devops' doesn't align  
incentives and interests

software is creative



software is complex

software is not digging ditches

software is not running factories

software is closer to art than science

Principles > Practices > Tools

mindset > skillset > toolset

adapt > adopt

why > what



smart motivated people  
working together

‘the best methodology’

# Call to Action

- be smart
- be motivated
- work together
- change your behavior
- change your behavior
- change your behavior

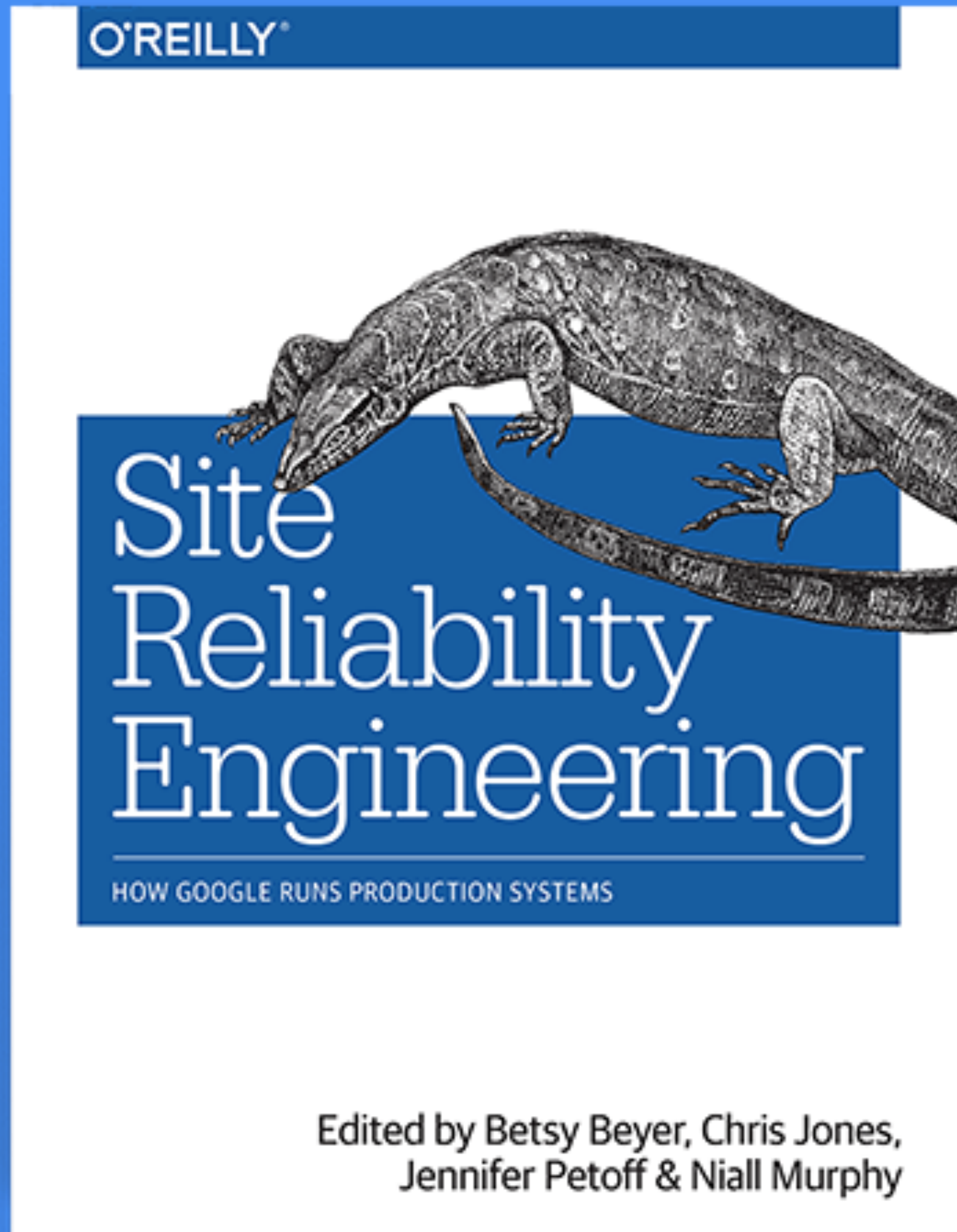


# BONUS

# Homework

- Embracing Risk
- Service Level Objectives
- Eliminating Toil

Bonus: Communication and Collaboration in SRE



# Site Reliability Engineering

Edited by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy

Members of the SRE team explain how their engagement with the entire software lifecycle has enabled Google to build, deploy, monitor, and maintain some of the largest software systems in the world.

[READ ONLINE FOR FREE](#) ↗

[BUY FROM GOOGLE BOOKS](#) ↗

<https://landing.google.com/sre/book.html>

# Google's devops implementation

# SRE - calms



- culture



- automation



- lean



- metrics



- sharing



# 謝謝

I'm not here to answer questions.  
I'm here to have conversations.

---



@littleidea



# Pivotal™

We are uncovering better ways of developing software,  
by doing it and helping others do it