

Jesse Robbins

Cofounder, Opscode

@jesserobbins

jesse@opscode.com

Hacking Culture



OPSCODE Chef



Hacking Culture:

“...may you at least serve as an example to others...”



amazon.com

FIRE DEPT

**Operations at Web Scale
is the ability to consistently create and
deploy reliable software to an unreliable
platform that scales horizontally.**

O'REILLY

Velocity

Web Performance
and Operations
Conference



O'REILLY

O'REILLY

Velocity

Web Performance and
Operations Conference



**Google, Amazon, Microsoft, Yahoo
built their own automation &
deployment tools**



but it was “secret sauce”



FAIL

everyone else was here

... inexperienced & poorly equipped for the world they must now operate in.

failblog.org



OPSCODE

Cloud Infrastructure Automation



AUTOMATE



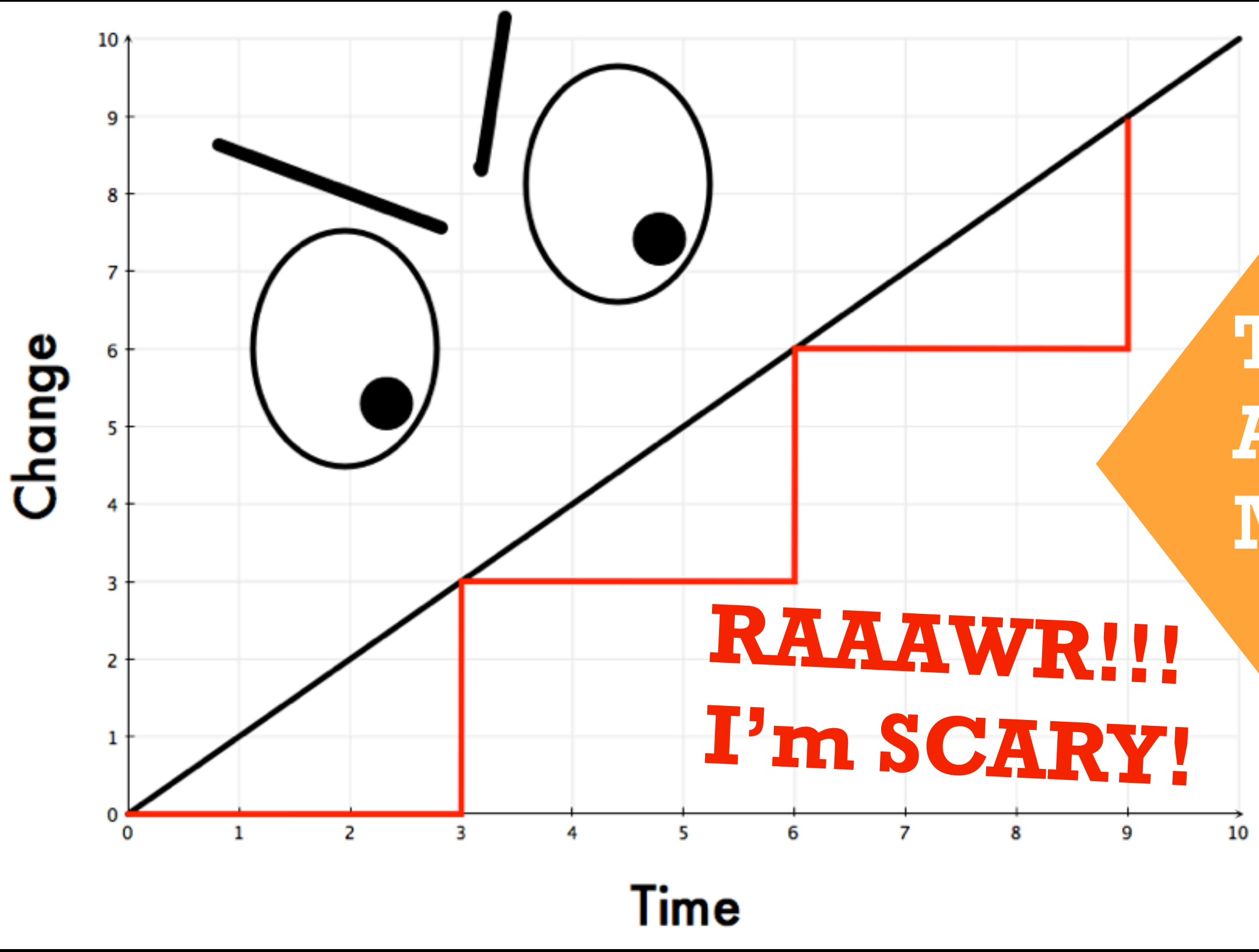
ALL THE THINGS!



Continuous Delivery

**This man is
John Allspaw**

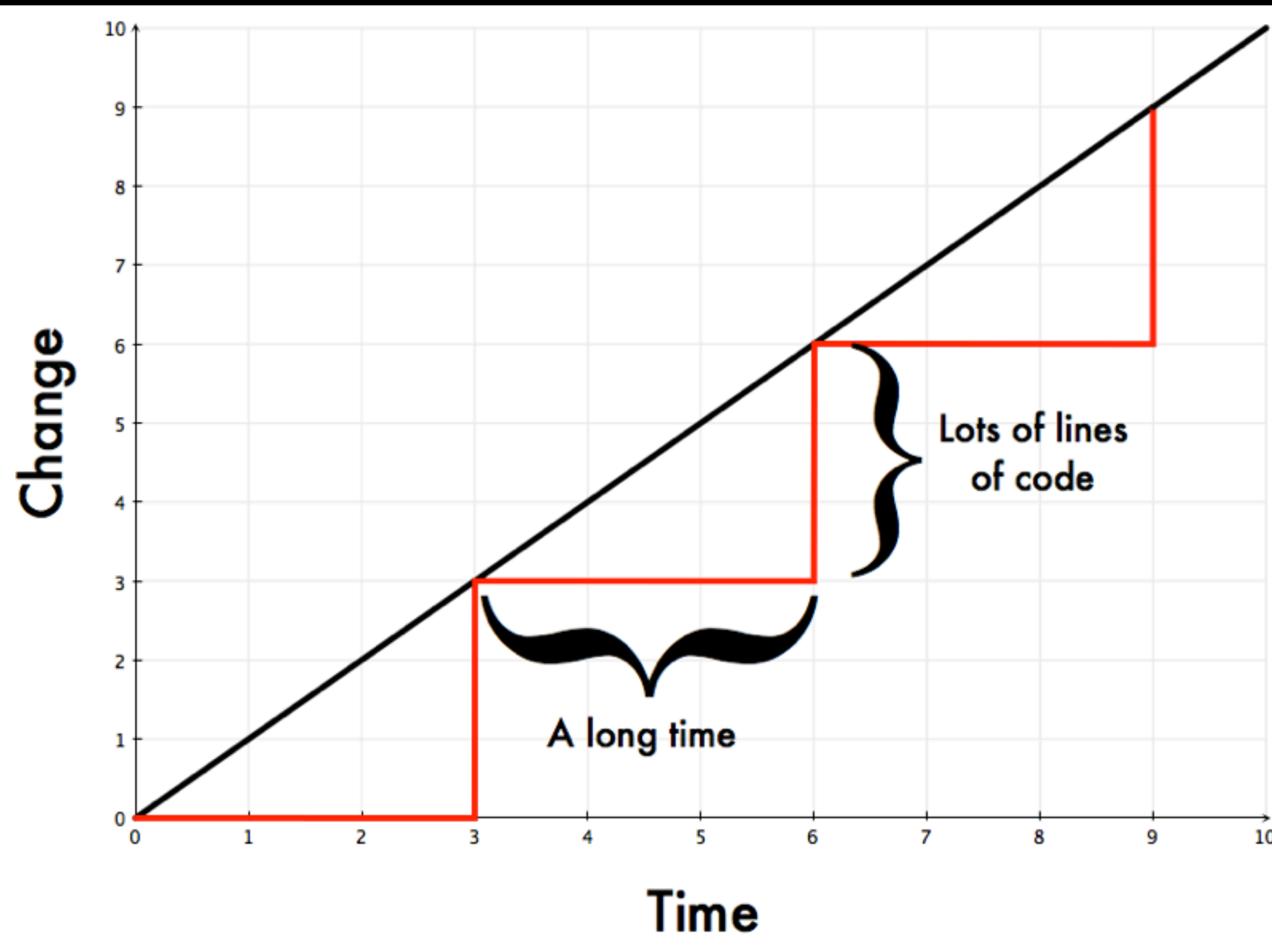
<http://www.flickr.com/photos/norby/7446208116/>

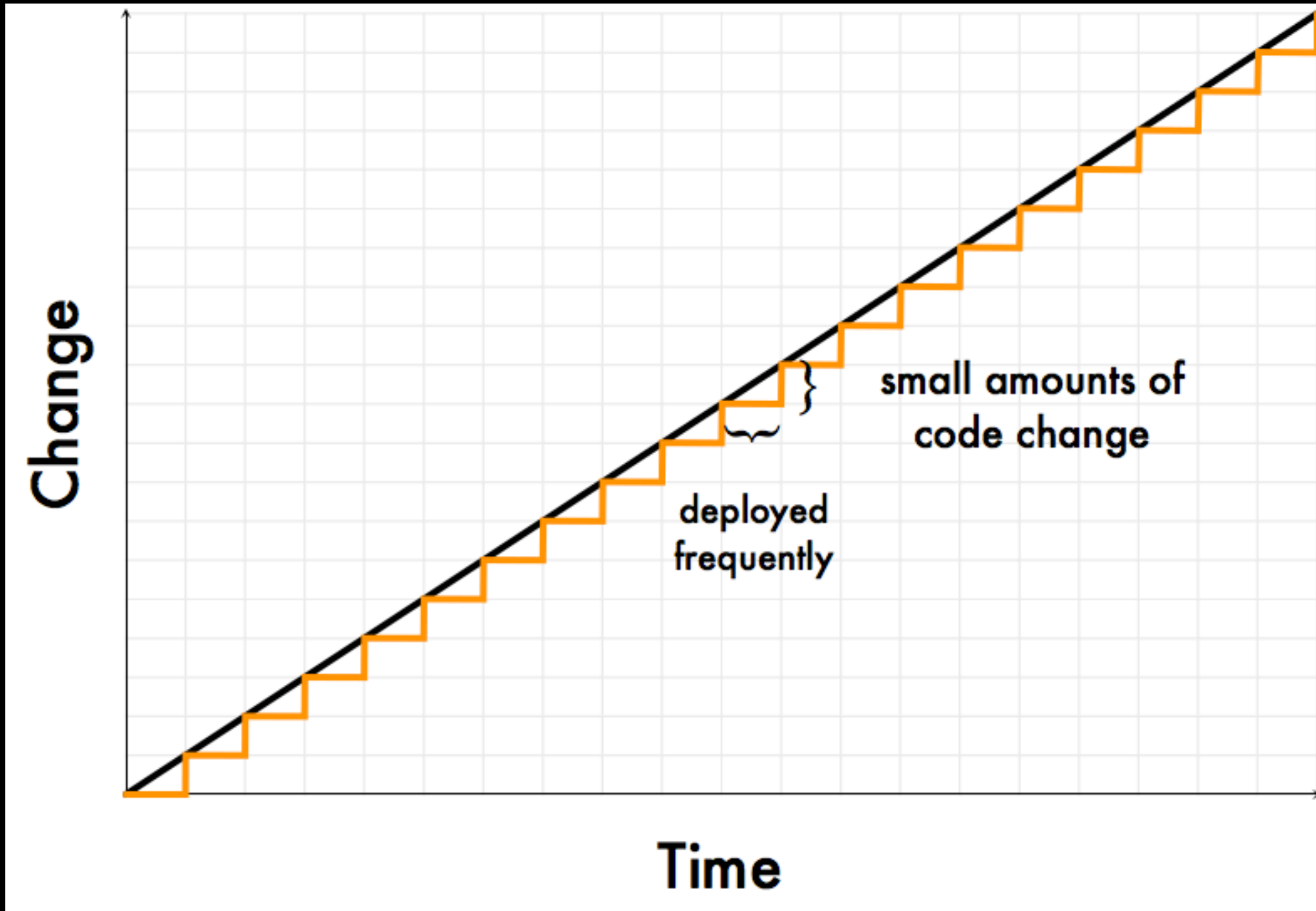


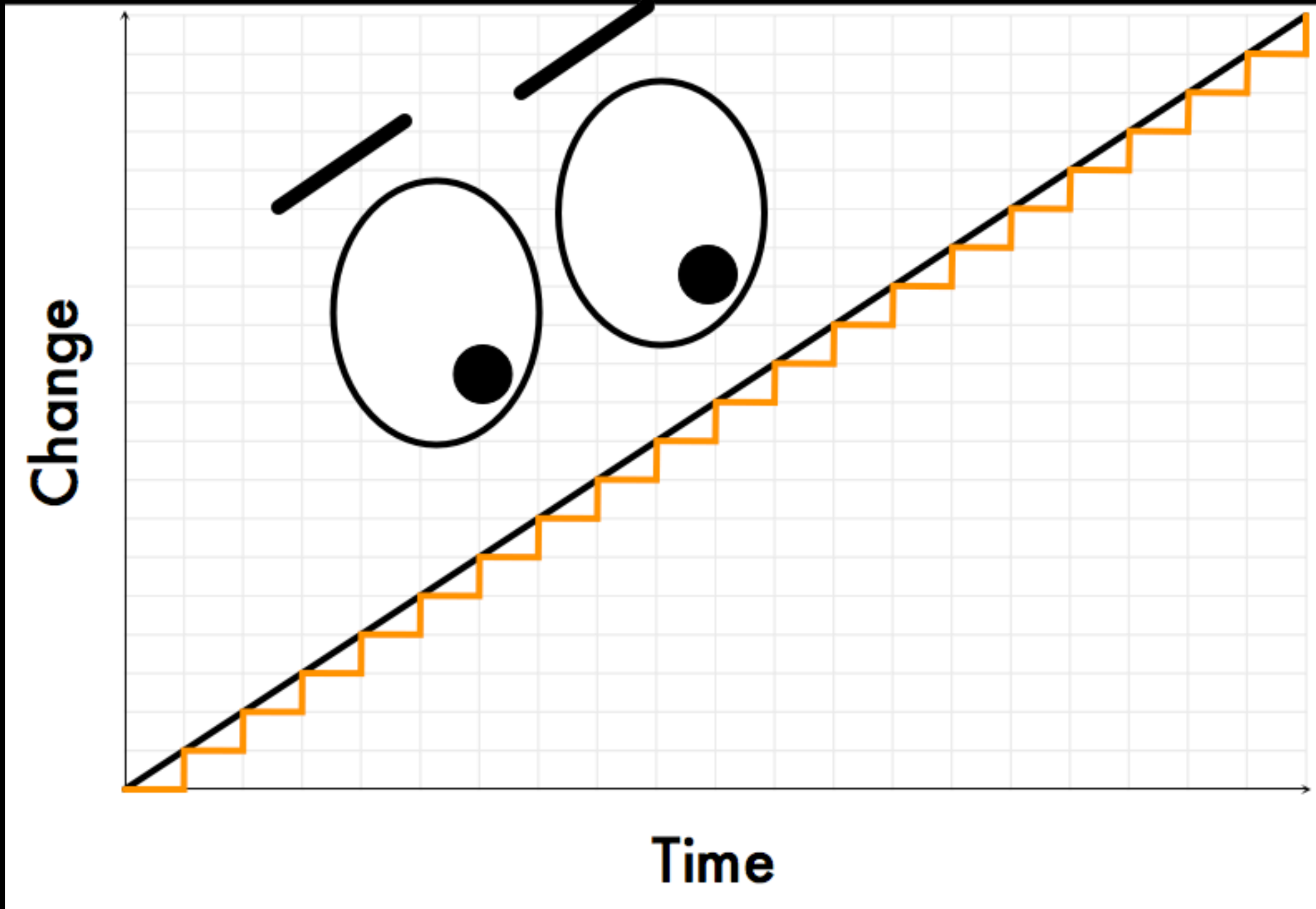
**This is
Allspaw's
Monster**

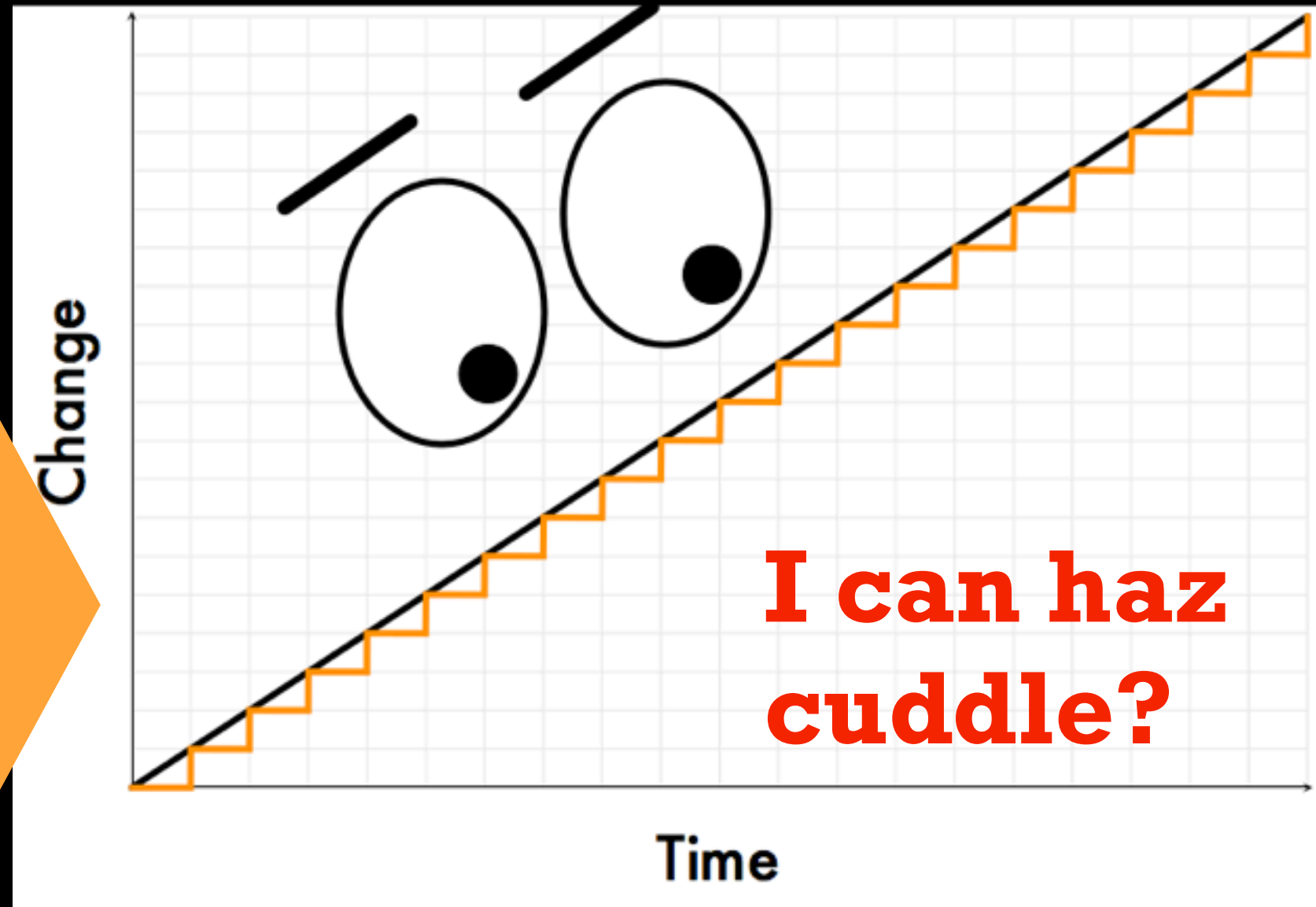
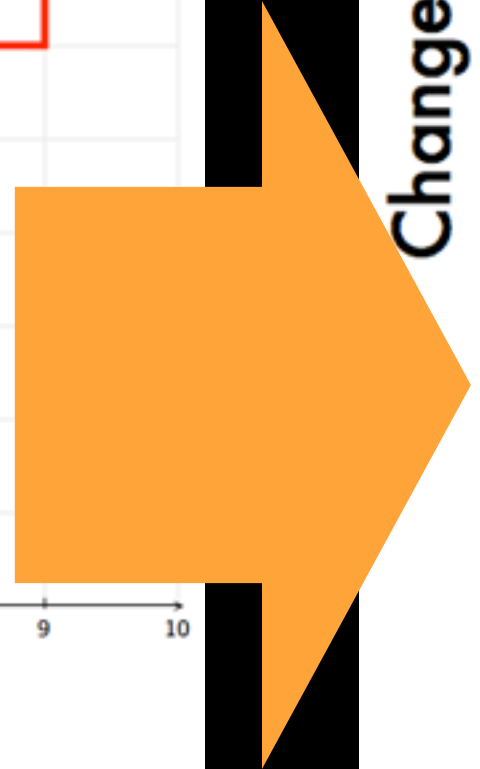
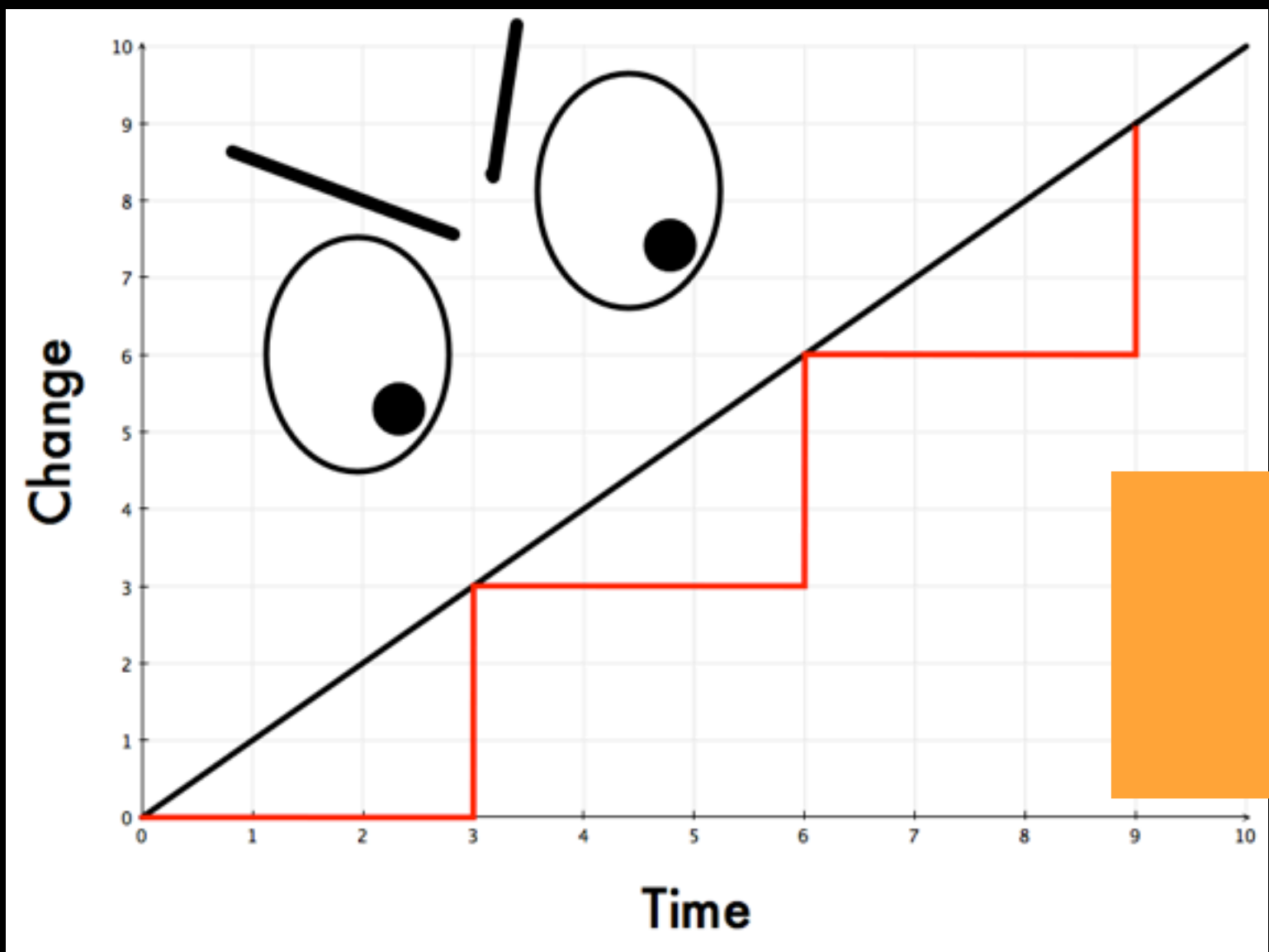


**RAAAWR!!!
I'm SCARY!**









Continuous Deployment means:

***Faster* Time to Value**

***Higher* Availability**

***Happier* Teams**

***More* Cool Stuff**

**Continuously Deploying code
*before you even write it.***



meanwhile... back at the office...

IT WORKED FINE IN TEST

OPS PROBLEM NOW

DONT MIND ME

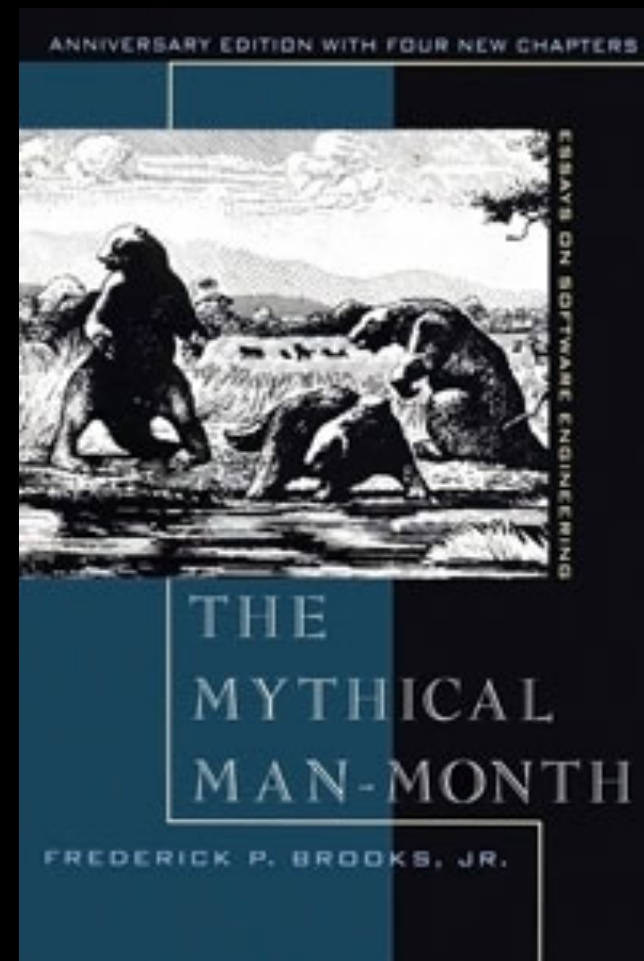
JUST SUPPORTING MY DEVELOPERS

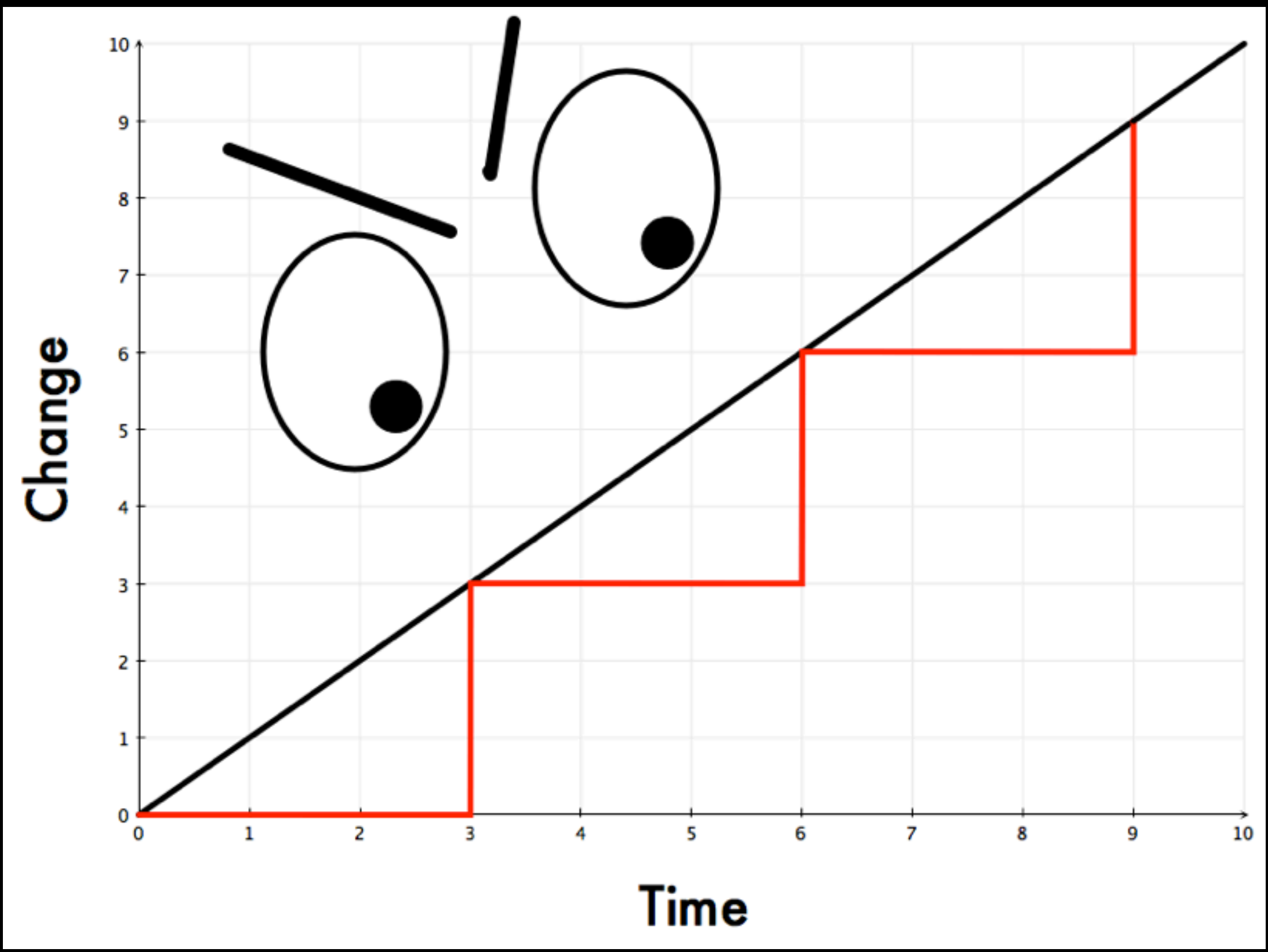
Tools* are not enough
(* even really great tools like Chef!)

Conway's law:

***“Organizations which design systems ...
are constrained to produce designs
which are copies of the communication
structures of these organizations...”***

Brook's said:
***“Quality is strongly affected by
organization structure.”***





Choose:

**Discourage change in the
interests of stability**

or

**Allow change to happen as often
as it needs to**

Common Attributes of Web Scale Cultures

Infrastructure *as Code*

- ▶ Full Stack Automation
- ▶ Commodity Hardware
- ▶ Reliability in software stack
- ▶ Datacenter APIs
- ▶ Core Infra Services
 - ▶ *Infrastructure as Product*
 - ▶ *App as Customer*

Application *as Services*


- ▶ Service Orientation
- ▶ Versioned APIs
- ▶ Software Resiliency (Design for Failure)
- ▶ Database/Storage Abstraction
- ▶ Complexity pushed up the stack
- ▶ Deep Instrumentation

Dev / Ops *as Teams*

- ▶ Shared Metrics / Monitoring
- ▶ Incident Management
- ▶ Service Owners On-call
- ▶ Tight integration
- ▶ Continuous Integration
- ▶ Continuous Deployment
- ▶ SRE/SRO
- ▶ GameDay

This isn't new

- ▶ **Theory of Constraints**
- ▶ **Lean / JIT**
- ▶ **Six Sigma**
- ▶ **Toyota Production System**
- ▶ **Agile**
- ▶ **etc...**

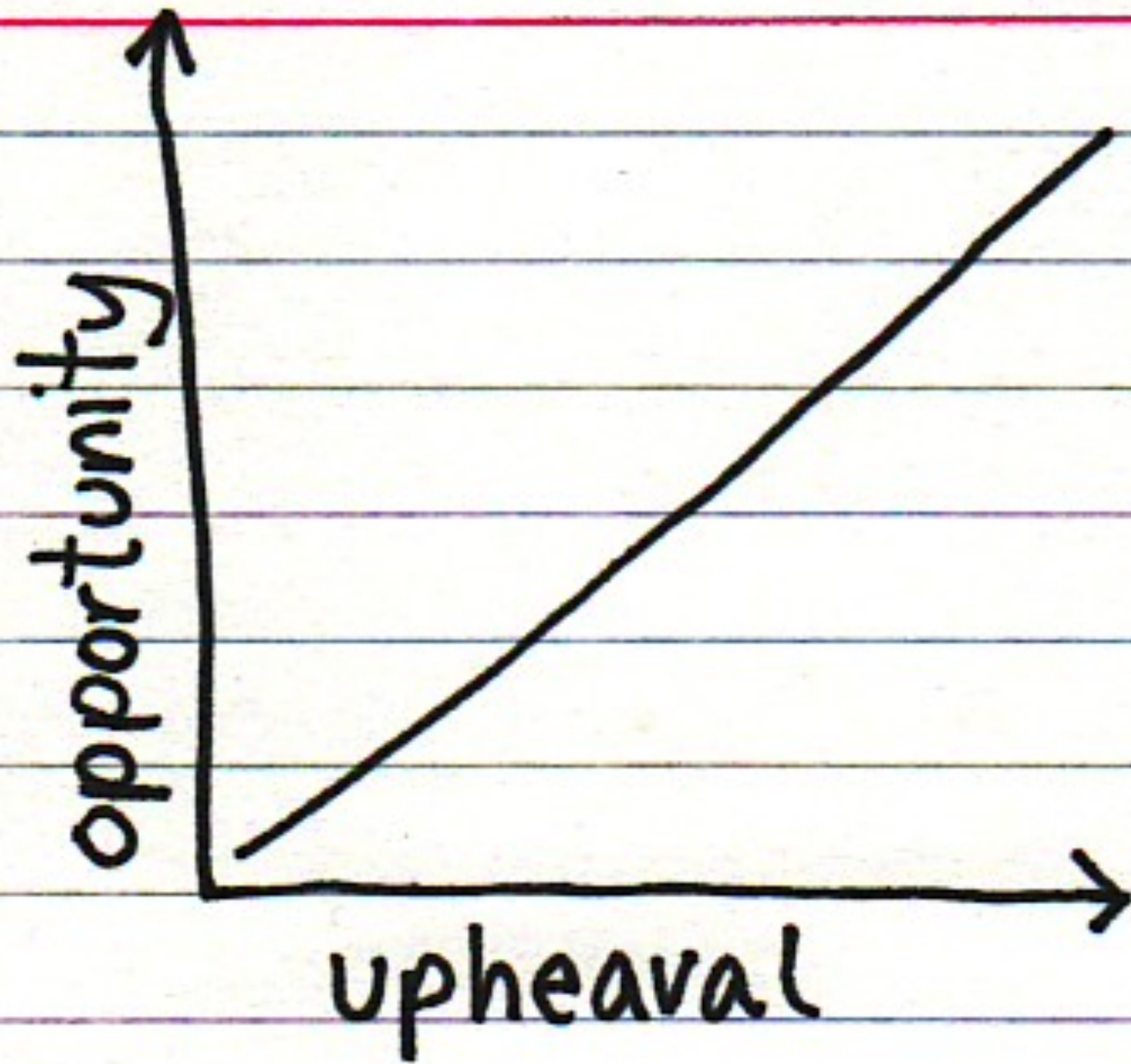
A close-up photograph of an elephant's head and large ear. The elephant's skin is dark grey and wrinkled. The ear is large and light brown. The background is a blurred natural setting with green foliage. A black speech bubble with white text is positioned in the upper right corner of the image.

***...but we can't
do it that way
because...***

**elephants cannot fly just by
flapping their ears harder...**

A photograph of a white egret standing on the back of an elephant's head. The egret is facing left, and the elephant's head is in the foreground, showing its eye and ear. The background is a dark, clear sky. A black text box is overlaid on the image.

To fly you must have wings, surface area, and a high power to weight ratio...



Common Attributes of Web Scale Cultures

Infrastructure *as Code*

- ▶ Full Stack Automation
- ▶ Commodity Hardware
- ▶ Reliability in software stack
- ▶ Datacenter APIs
- ▶ Core Infra Services
 - ▶ *Infrastructure as Product*
 - ▶ *App as Customer*

Application *as Services*

- ▶ Service Orientation
- ▶ Versioned APIs
- ▶ Software Resiliency (Design for Failure)
- ▶ Database/Storage Abstraction
- ▶ Complexity pushed up the stack
- ▶ Deep Instrumentation

Dev / Ops *as Teams*

- ▶ Shared Metrics / Monitoring
- ▶ Incident Management
- ▶ Service Owners On-call
- ▶ Tight integration
- ▶ Continuous Integration & Deployment
- ▶ SRE/SRO
- ▶ GameDay

**You cannot change everyone or
everything at once.**

Change takes time
(sorry about that)

**Choose your battles, or better
yet, don't battle at all...**

Jesse's Rule:

***Don't Fight Stupid,
Make More Awesome!***

Changing Culture:

- 1. Start small, build trust & safety**
- 2. Create Champions**
- 3. Use metrics to build confidence**
- 4. Celebrate successes**
- 5. Exploit Compelling Events**

Changing Culture:

- 1. Start small, build trust & safety**
- 2. Create Champions**
- 3. Use metrics to build confidence**
- 4. Celebrate successes**
- 5. Exploit Compelling Events**

Hacks: Starting Small

- 1. Small isn't a threat & is easy to ignore**
- 2. Just call it an experiment**

Hacks: Creating Champions

- 1. Get executive sponsors, starting with your boss.**
- 2. Give everyone else the credit.**
- 3. Give “Special Status”**
- 4. Have people with “Special Status” talk about the new awesome.**

Hacks: Metrics

- 1. Find KPI that supports change**
- 2. Track and use it ruthlessly - first to show value, later cost of not making the change by laggards**
- 3. Tell your story with data**

Hacks: Celebrating successes

- 1. Tell a powerful story**
- 2. Always positive about people and how they overcame a problem.**
- 3. Never about people who created the problem.**
- 4. Leave room for people to come to your side. (*don't fight stupid ;-)*)**

Hacks: Compelling Events

- 1. Just wait, it will come**
- 2. Can be created by things like compliance, scaling, cloud migrations**
- 3. Not “I told you so” - but “what do we do now”**



Hacking Tools

Infrastructure as Code:

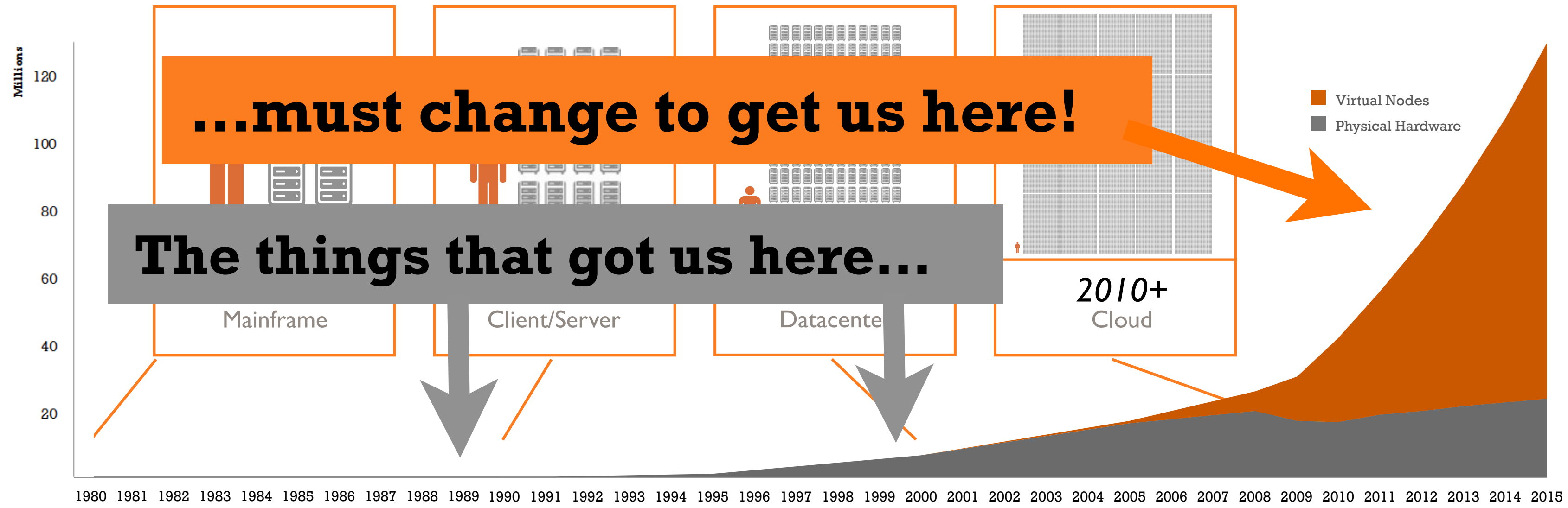
Enable the reconstruction of the business from nothing but a source code repository, an application data backup, and bare resources.



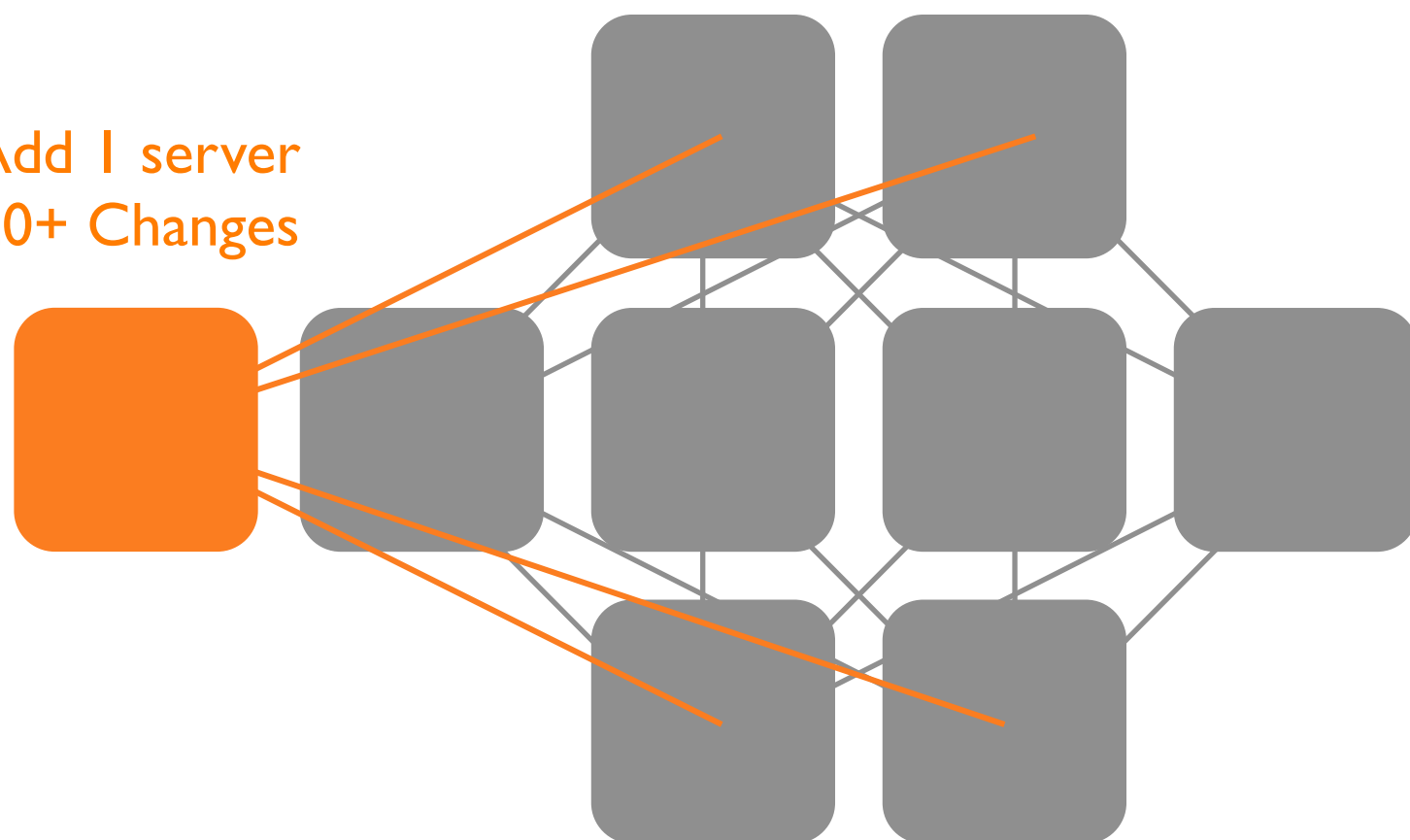
OPSCODE



Chef



Add 1 server
20+ Changes



Frontend Web
Servers

Application Servers

Database

To Add a New Server...

2x Web Server
Configurations

2 Web Server Restarts

4x Database Configurations

8x Firewall Configurations

DNS Service

Network Configuration

Deployer

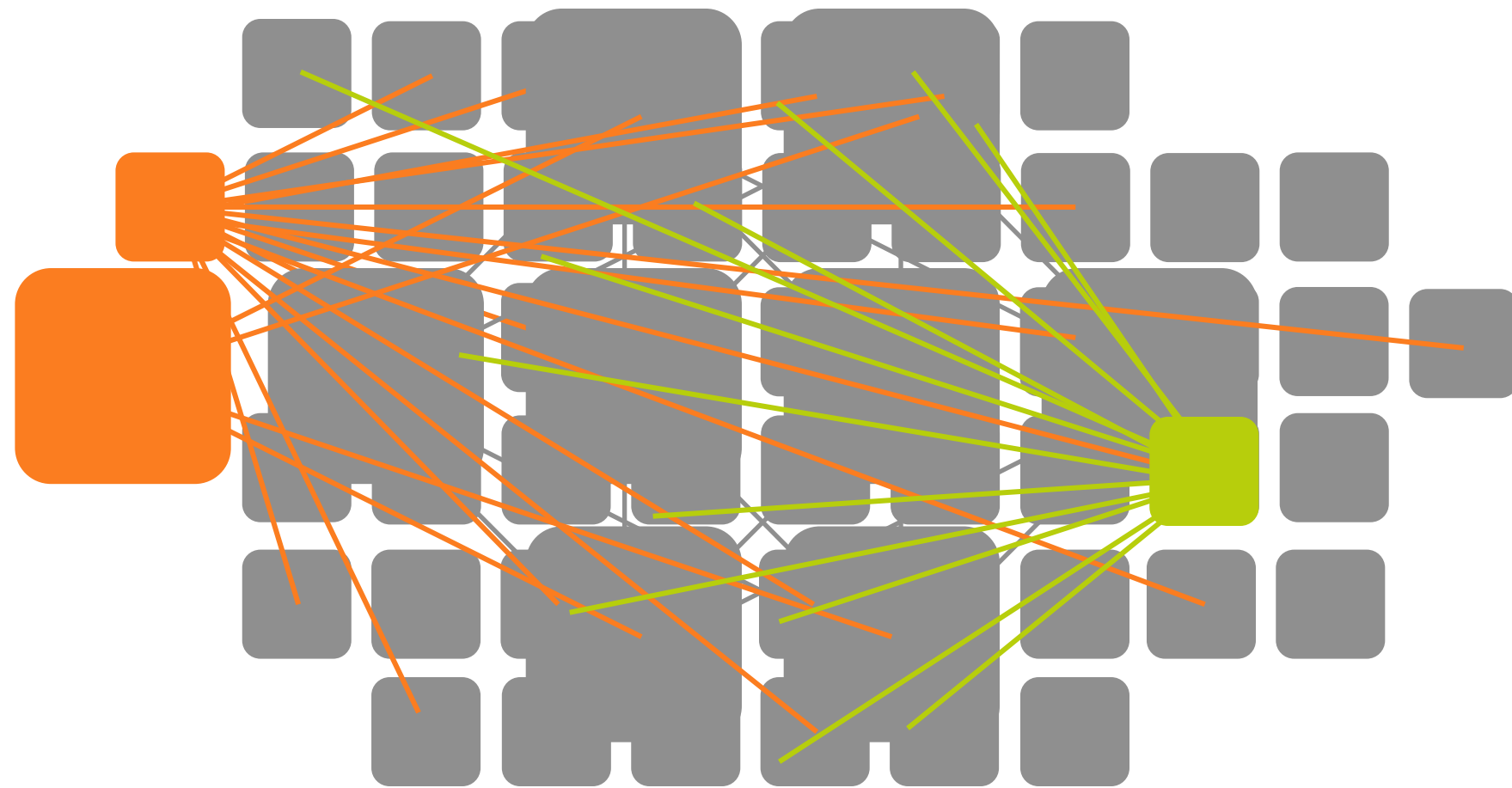
8x Monitoring Changes

The Bottom Line...

20+ Changes

12+ New Infrastructure
Dependencies

4+ Hours



We added:

Load Balancers

MemCache

Search Appliances

Lots of VM's

More Scale

Exponential Increase In

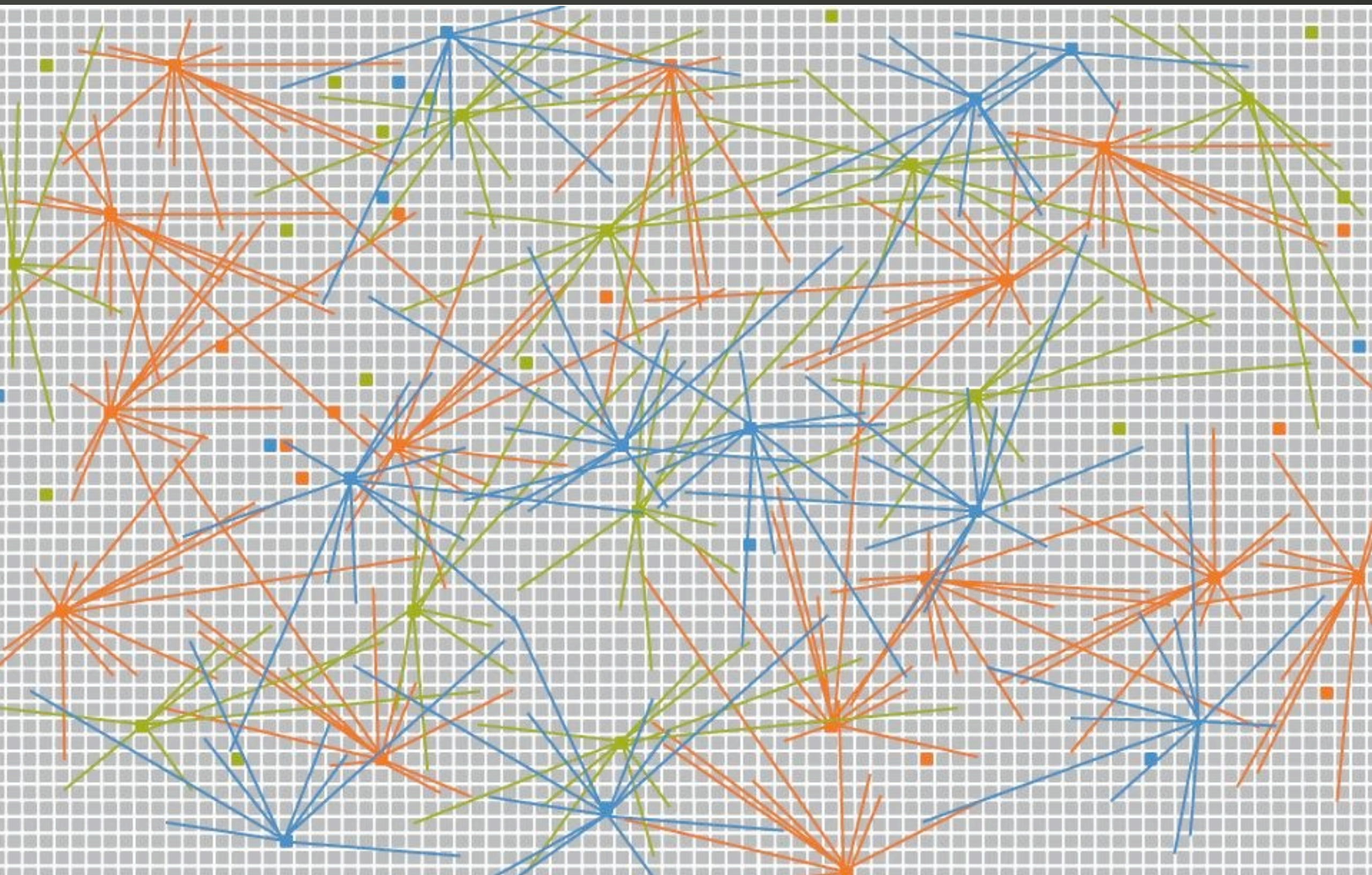
Configuration Changes

Infrastructure

Dependencies

Skills Needed

Greater Risk



How Do we Manage This at Cloud Scale?

Thousands of infrastructure dependencies and configurations needed for each change.

Huge Amounts of Time

Increased Cost of Correction of Manual Errors

Huge Need for Talent

Risk of Critical Skills Shortage



AUTOMATE



ALL THE THINGS!



Infrastructure as Code:

Enable the reconstruction of the business from nothing but a source code repository, an application data backup, and bare resources.



OPSCODE



Chef



Discovery and
Visibility

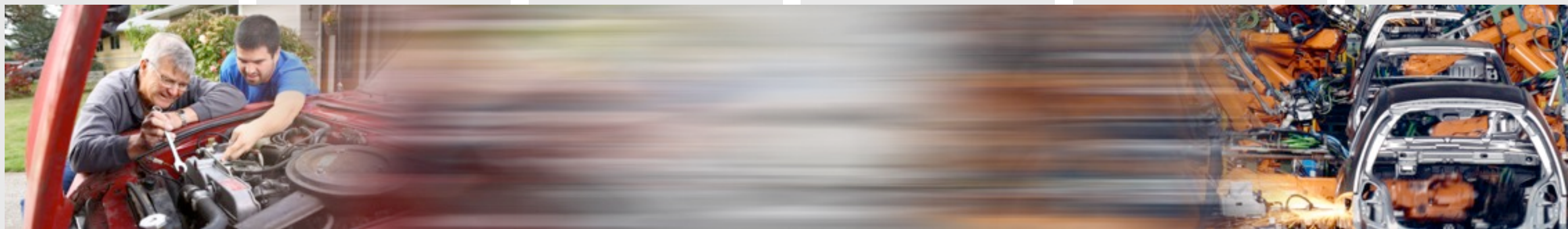
Common
Automation Tasks:
Scripts, OS
Compliance,
Updates & Patches

Configuration
Management

Application
Management

Continuous
Deployment

Full
Infrastructure
Automation



Hacking Permission

**Operation
Hurricane Katrina**



1619

JESSE G ROBBINS

HANCOCK CO, MS

MASS CARE

ESF-6

FULL ACCESS

FEMIA: “Who is in charge here”

Anarchists: “NOBODY!”

**FEMIA: “Okay, you have to shut
down so we can feed people”**

**Anarchists: “We’re feeding 5000
people a day.”**

**Solution: Make all volunteers
“site directors”**

FEMIA: “Who is in charge here?”

Anarchist: “I’m a site director...”

FEMIA: “Please sign for your supplies”



Jesse's Rule:

***Don't Fight Stupid,
Make More Awesome!***

Jesse Robbins

Cofounder, Opscode

@jesserobbins

jesse@opscode.com



OPSCODE Chef

