iZettle

public class SRE implements DevOps

iZettle

Tech Away Day Nov 2018

Who am I?

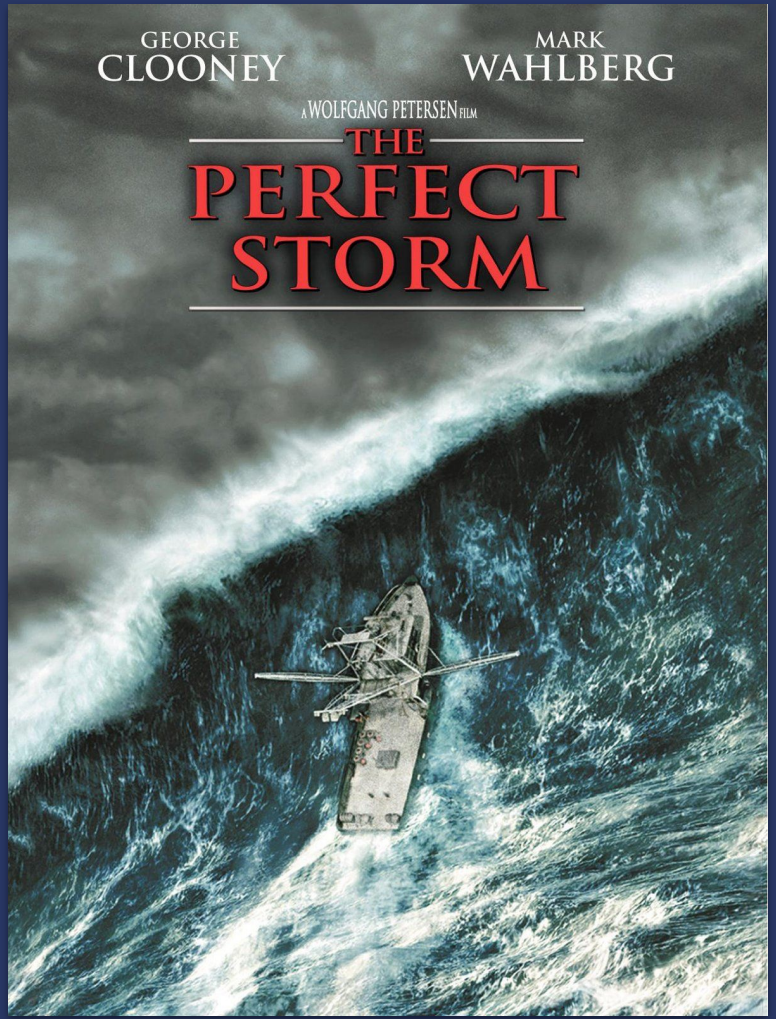
Who am I?

- Have come from more of an Ops background
- Am really interested in how and why systems breaks
- Am interested in how people respond to incidents

Why am I giving this talk?

Over the next 12 months we are going to have a lot of incidents

iZettle


GEORGE
CLOONEY
MARK
WAHLBERG
A WOLFGANG PETERSEN FILM
THE
PERFECT
STORM

iZettle

Roughly 70%[1] of outages are due to changes

[1] *Analysis of Google internal data 2011 - 2018*
Solving reliability fears with SRE Cloud Next '18

iZettle

How we manage those incidents will define our success

iZettle

SRE provides a clean outline of how to manage incidents and failure

iZettle

What is DevOps?

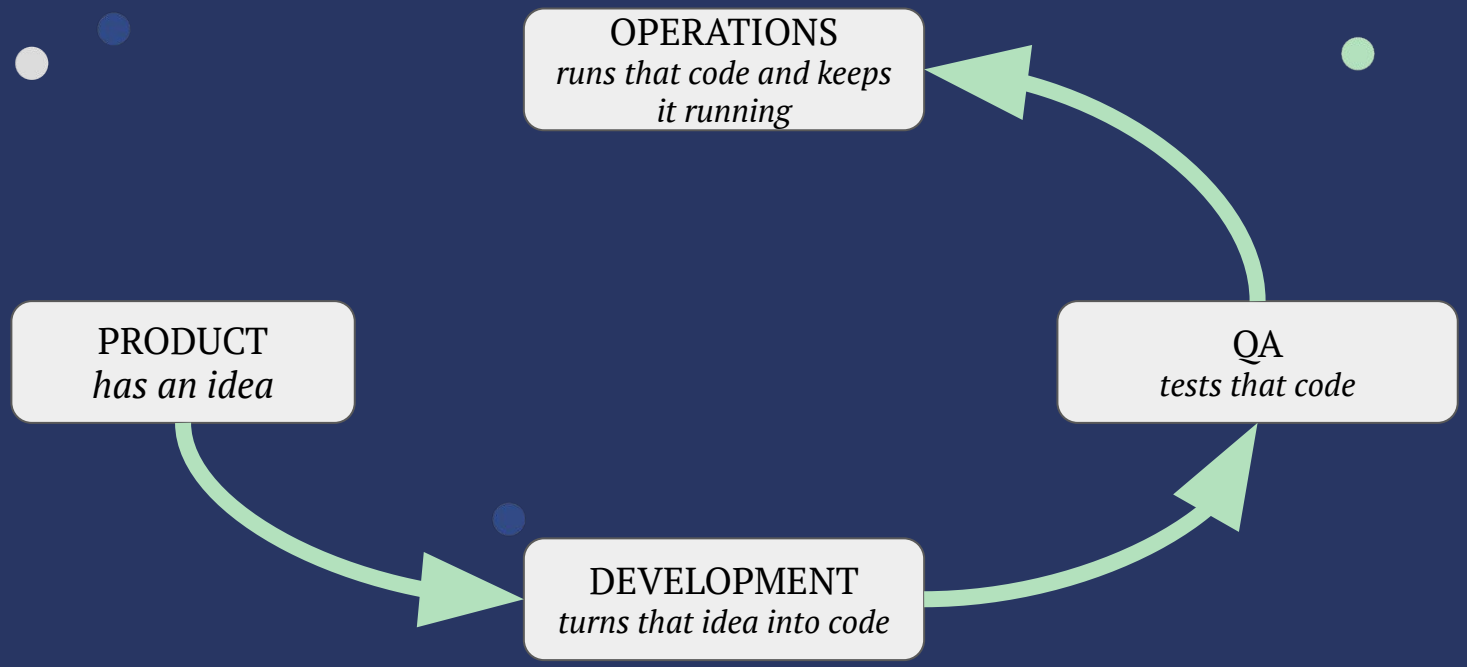
iZettle

Software Service Operations is hard

iZettle

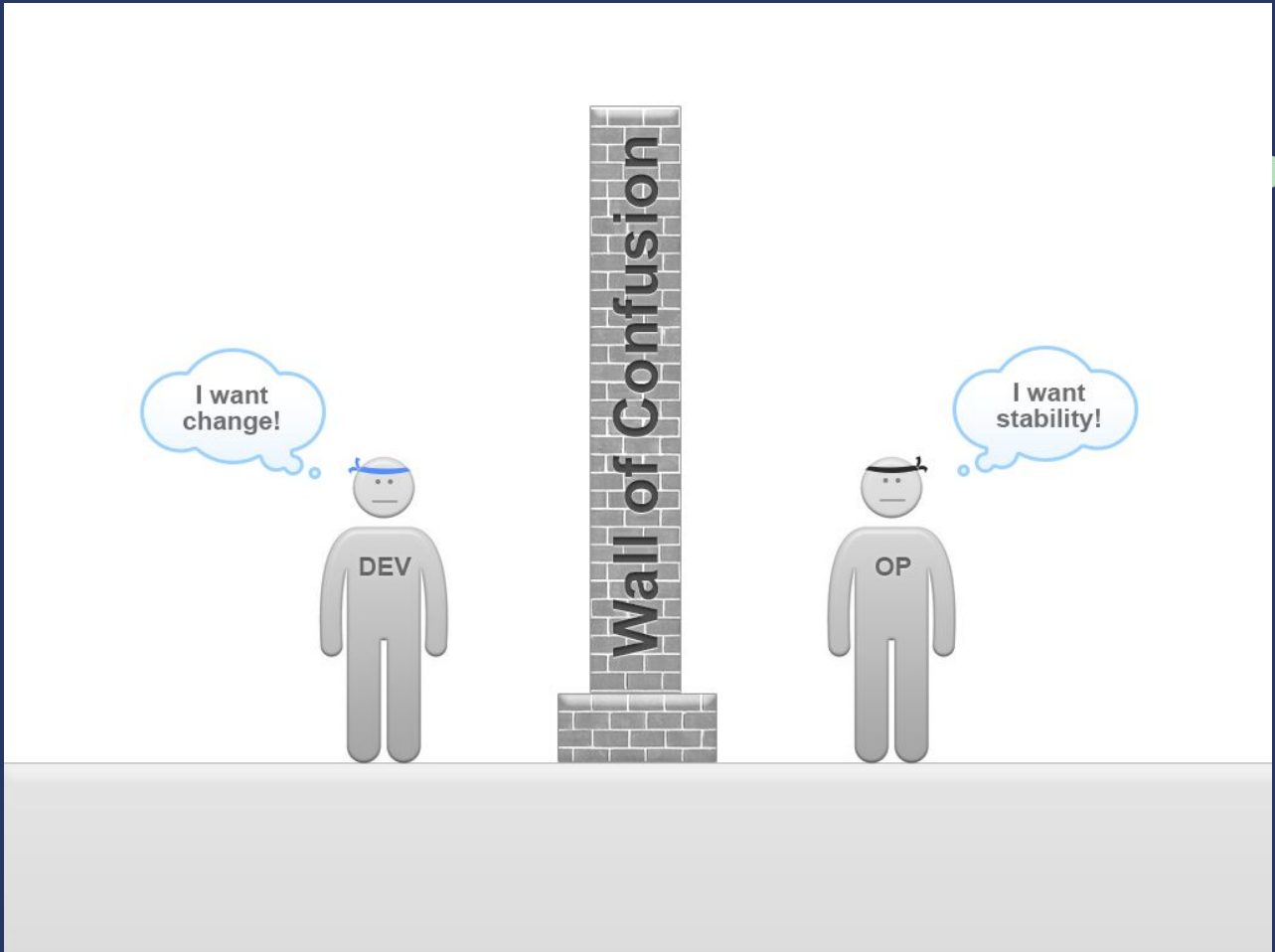
Traditionally Operations was treated as a Cost Centre

iZettle


OPERATIONS
runs that code and keeps it running
PRODUCT
has an idea
QA
tests that code
DEVELOPMENT
turns that idea into code

iZettle

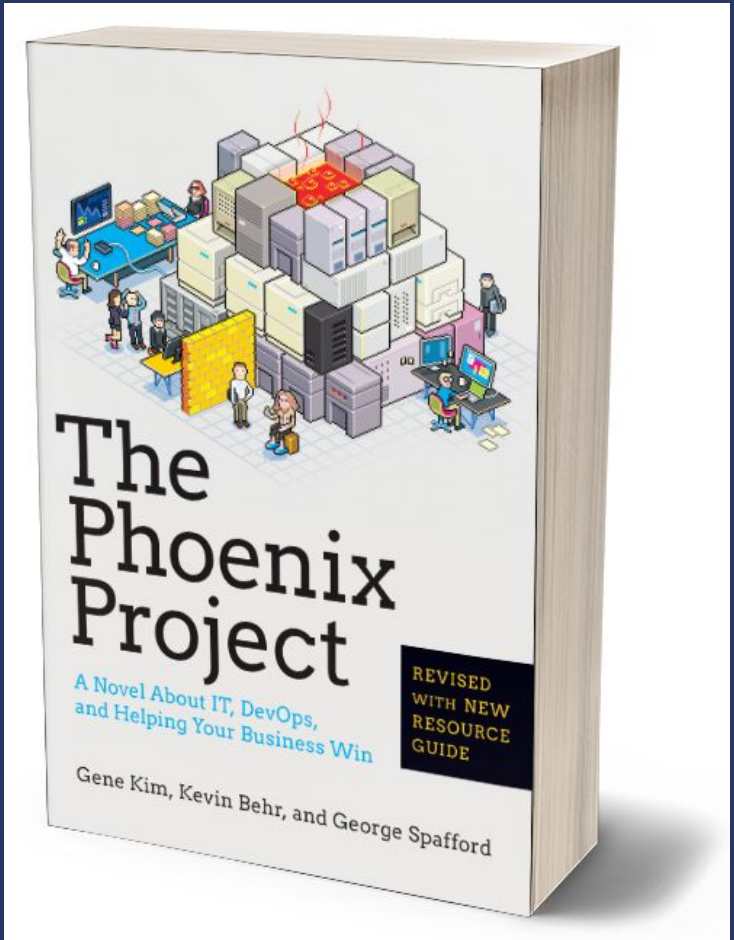
iZettle



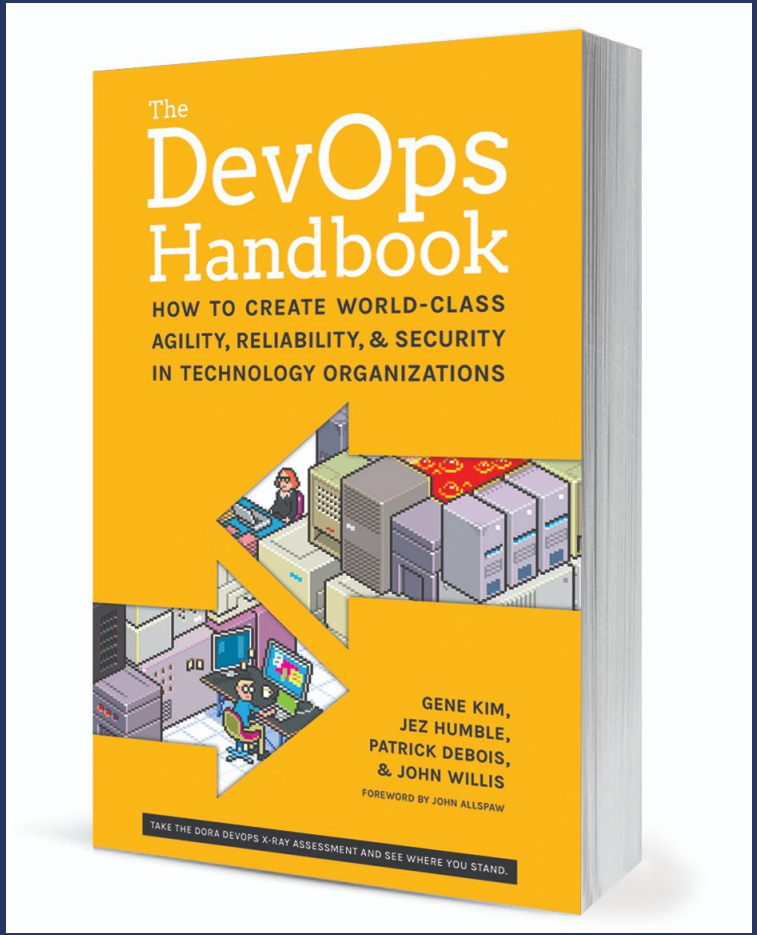
iZettle


REST ROOMS
DEV
OPS
Looks like they have a problem with silos.
Ya think?
ROELBOB 2016

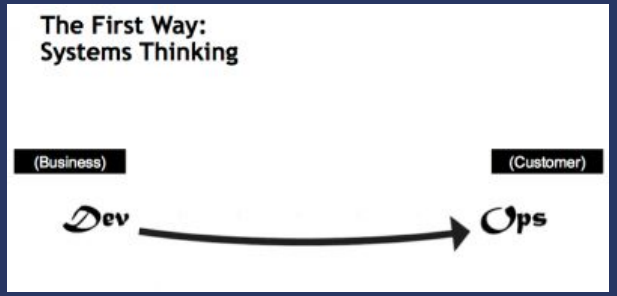
iZettle


The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
REVISED WITH NEW RESOURCE GUIDE
Gene Kim, Kevin Behr, and George Spafford

iZettle

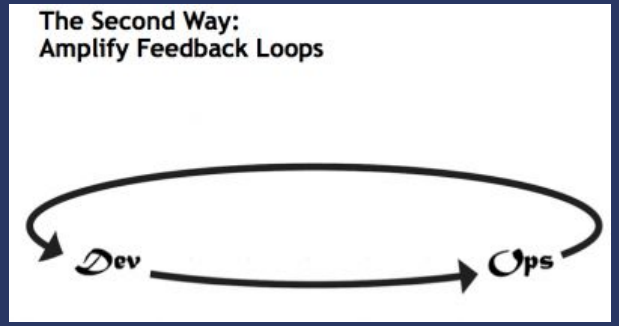

The
DevOps
Handbook
HOW TO CREATE WORLD-CLASS
AGILITY, RELIABILITY, & SECURITY
IN TECHNOLOGY ORGANIZATIONS
GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS
FOREWORD BY JOHN ALLSPAW
TAKE THE DORA DEVOPS X-RAY ASSESSMENT AND SEE WHERE YOU STAND.

iZettle

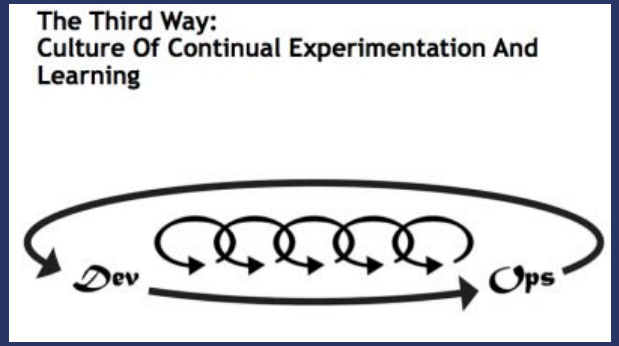


The First Way emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department..

The focus is on all business value streams that are enabled by IT.

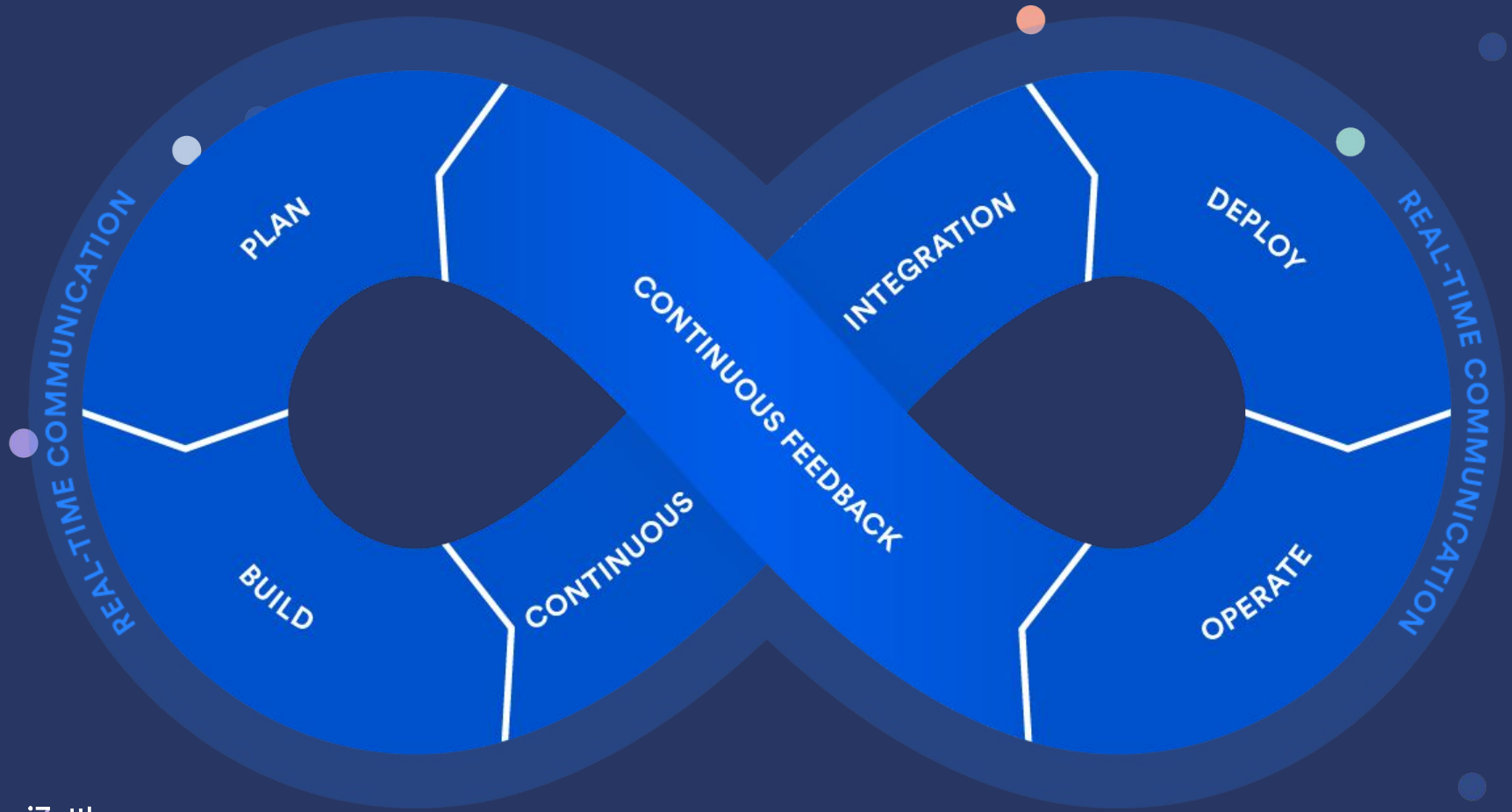


The Second Way is about creating the right to left feedback loops. The goal of almost any process improvement initiative is to shorten and amplify feedback loops so necessary corrections can be continually made.



The Third Way is about creating a culture that fosters two things: continual experimentation, taking risks and learning from failure; and understanding that repetition and practice is the prerequisite to mastery.

We need both of these equally. Experimentation and taking risks are what ensures that we keep pushing to improve, even if it means going deeper into the danger zone than we've ever gone. And we need mastery of the skills that can help us retreat out of the danger zone when we've gone too far.


PLAN
BUILD
CONTINUOUS
INTEGRATION
DEPLOY
OPERATE
CONTINUOUS FEEDBACK
REAL-TIME COMMUNICATION
REAL-TIME COMMUNICATION

iZettle

ACCEPT FAILURE AS NORMAL

IMPLEMENT GRADUAL CHANGE

REDUCE ORGANISATIONAL SILOS

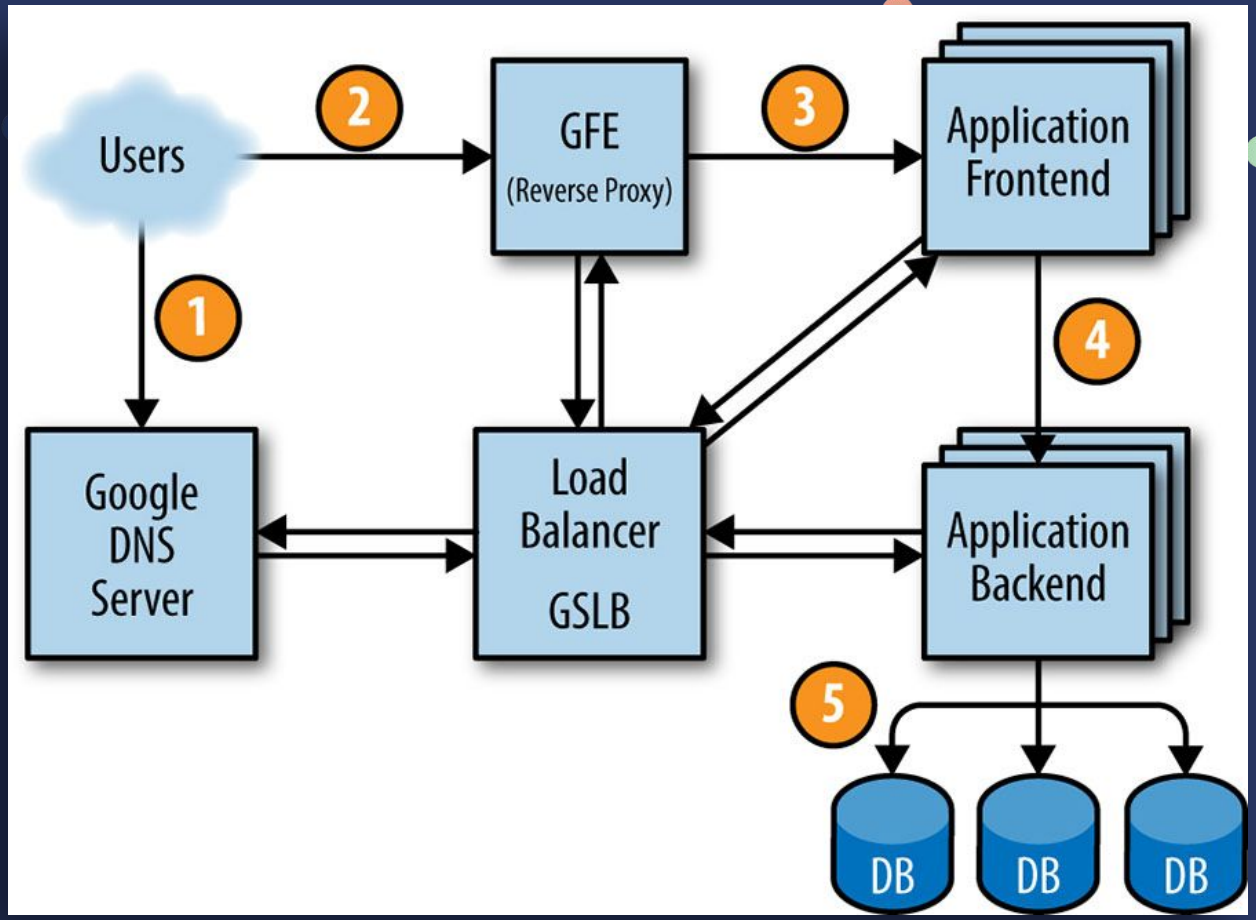
LEVERAGE AUTOMATION

MEASURE EVERYTHING

iZettle

What about SRE?

History


Users
2
GFE
(Reverse Proxy)
3
Application
Frontend
1
4
Google
DNS
Server
Load
Balancer
GSLB
Application
Backend
5
DB
DB
DB

iZettle

SRE Principles

- Use data to guide decisions
- Embracing Risk through Error Budgets
- Service Level Objectives that have consequences
- Eliminating Toil so we have time to make tomorrow better

Use Data to Guide Decisions

iZettle

Embracing Risk

iZettle

100% uptime is not feasible

iZettle

Most laptops are ~70% reliable

iZettle

The difference between 99.9% and 99.999% is indistinguishable for most users

iZettle

Reliability and Innovation and competing goals

iZettle

Error Budget

iZettle

Error Budget

- Given an SLO of 99.9% what downtime can we have?

$$\frac{100 - 99.9}{100} \times \underset{\text{days / year}}{365} \times \underset{\text{hours / day}}{24} \times \underset{\text{mins / hour}}{60} = \underset{\text{mins / year}}{525}$$

Availability Level	Allowed Unavailability Window		
	per year	per quarter	per 30 days
90%	36.5 days	9 days	3 days
95%	18.25 days	4.5 days	1.5 days
99%	3.65 days	21.6 hours	7.2 hours
99.5%	1.83 days	10.6 hours	3.6 hours
99.9%	8.76 hours	2.16 hours	43.2 minutes
99.95%	4.38 hours	1.08 hours	21.6 minutes
99.99%	52.6 minutes	12.96 minutes	4.32 minutes
99.999%	5.26 minutes	1.30 minutes	25.9 seconds

iZettle

Error Budget

- Given an SLO of 99.9%, after a 20 minute outage you still have 23 minutes of budget available for the remainder of the month.

Exceeding Your Error Budget

- Deployment of new features not allowed
- Focus on reliability/performance improvements

iZettle

Service Level **Objectives**
Service Level **Indicators**
Service Level **Agreements**

iZettle

Service Level Indicator

- Something you can measure
  - Request latency
  - Error rates

- *95th Percentile latency of homepage requests over past 5 minutes less than 300ms*

Service Level Objective

- Informed by SLIs
- Target level or range that states if a service is normal
- Specifically tracking customer experience
- If SLO is being met then customers are happy

- *95th percentile homepage latency SLI will succeed 99.9% of the time within a month*

Service Level Agreement

- Contractual obligation
- If breached then need to pay someone **£££**

- *Users will receive service credits if 95th percentile homepage latency SLI succeeds less than 99.9% over the trailing month*

SLIs drive **SLOs** which inform **SLAs**

iZettle

Error Budgets and SLOs

- Common incentive for Dev and Ops
- Dev teams decide how they want to spend their budget
- Unrealistic reliability goals become unattractive
- Builds shared responsibility between teams

Eliminating Toil

iZettle

Toil

- SRE puts a cap of 50% on the aggregate “ops” work.
  - Being on-call
  - Tickets
  - Manual tasks

Toil

- Toil is work that is:
  - **Manual**
  - **Repetitive**
  - **Automatable**
  - Devoid of **long-term value**

Toil

- What toil is not but is overhead
  - **Meetings**
  - **Emails**
  - **Travelling**
  - **Expenses**

Pursue Maximum Change Velocity Without Violating a Service's SLO

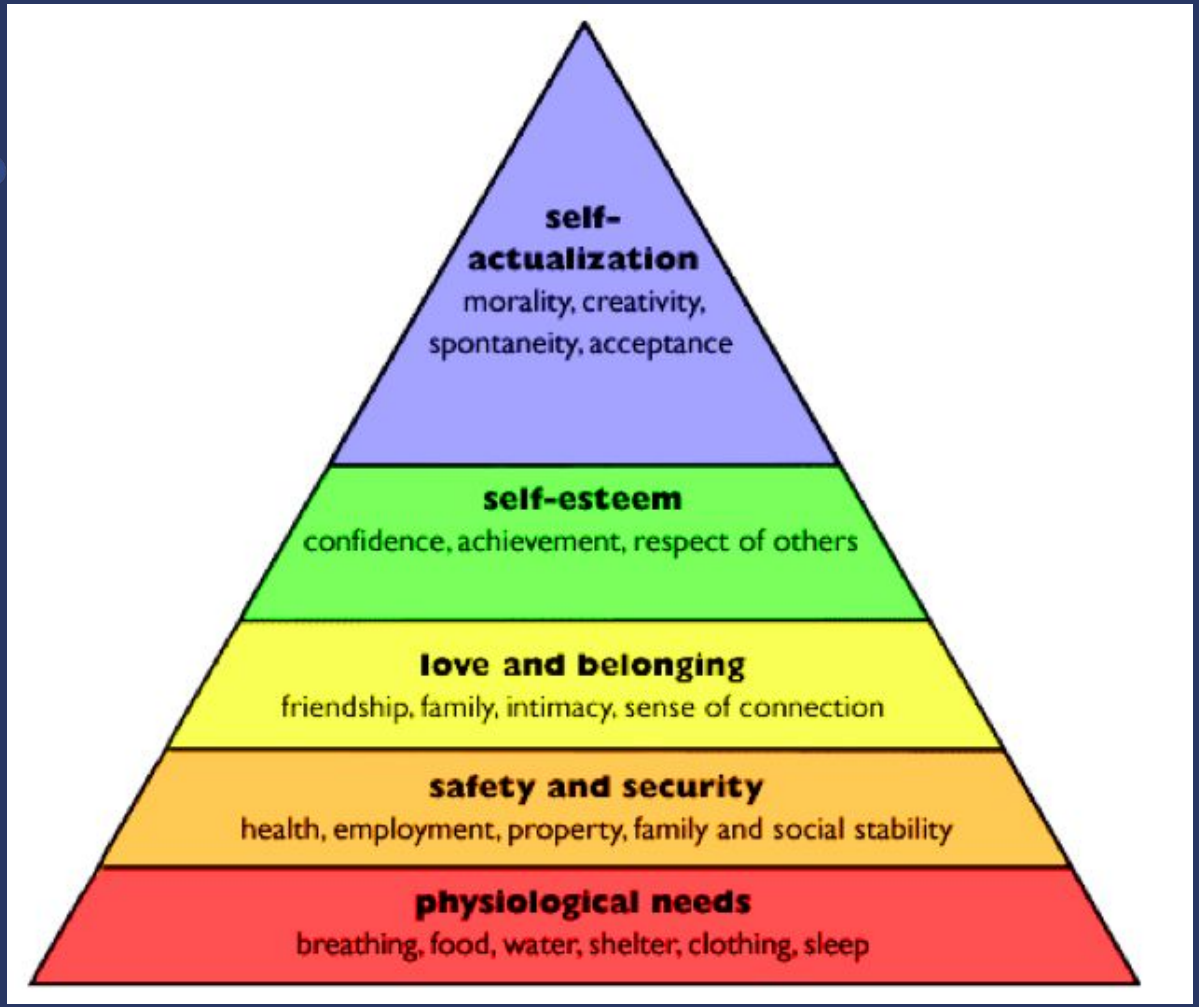
iZettle

Reducing Risky Behaviour with Canary Releases

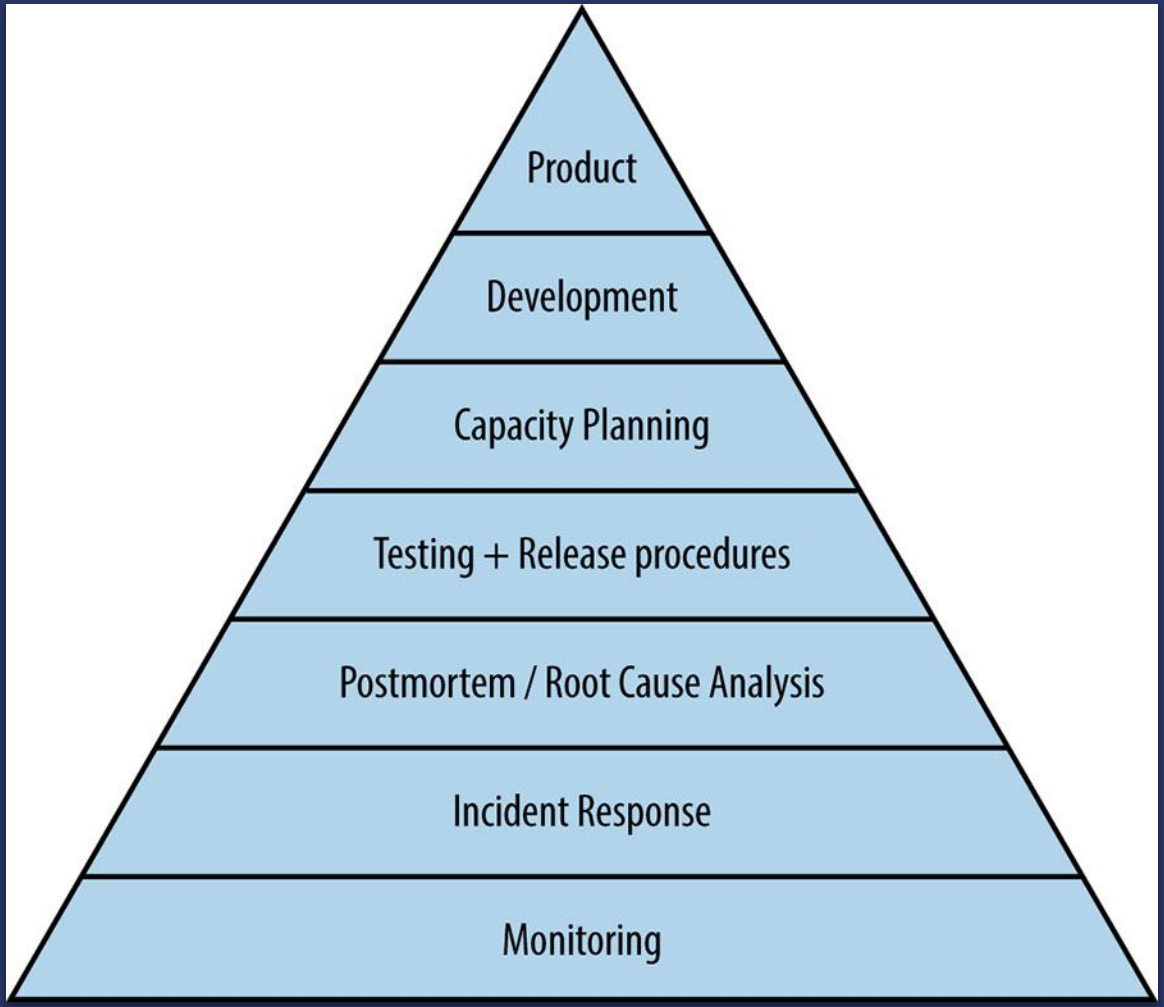
iZettle

The Hierarchy of Reliability

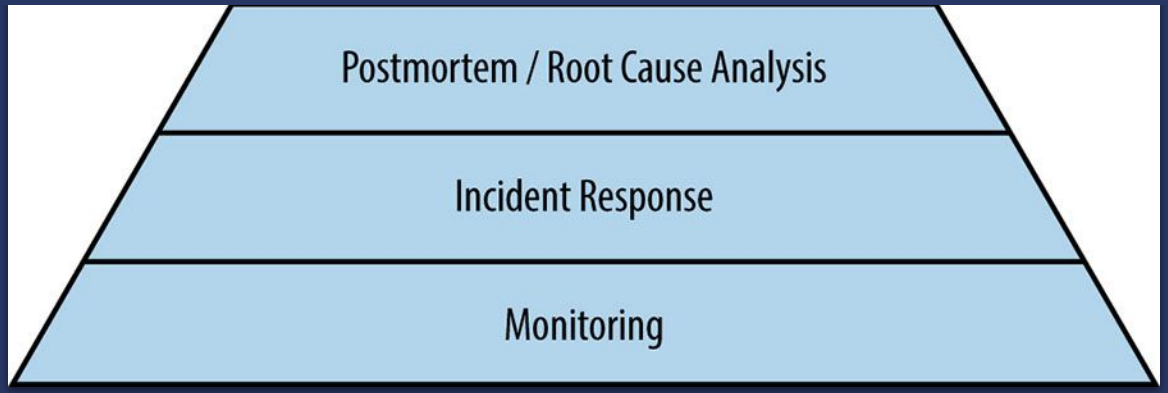
iZettle


self-
actualization
morality, creativity,
spontaneity, acceptance
self-esteem
confidence, achievement, respect of others
love and belonging
friendship, family, intimacy, sense of connection
safety and security
health, employment, property, family and social stability
physiological needs
breathing, food, water, shelter, clothing, sleep

iZettle


Product
Development
Capacity Planning
Testing + Release procedures
Postmortem / Root Cause Analysis
Incident Response
Monitoring

iZettle


Postmortem / Root Cause Analysis
Incident Response
Monitoring

iZettle

iZettle

Monitoring is necessary to:

- **Alert** on conditions that require attention
- **Investigate** and **diagnose** those issues
- Gain **insight** into **trends** in resource usage or service health for **long-term planning**
- **Compare** the behaviour of the system before and after a change, or between groups in an **experiment**

Data Sources

1. Logs

   Append-only record of events

2. Metrics

   Numerical measurements representing attributes and events, gathered at regular intervals

3. Distributed Traces

   Individual request level observability

Data Sources

1. Logs

   More granular, high cardinality but delayed delivery

2. Metrics

   Less granular, low cardinality, near real time and service level

3. Traces

   Very granular, high cardinality and focussed on individual requests

Monitoring is there to give you the context you need to fix failing services

iZettle

What should you observe?

The Four Golden Signals

1. **Latency**

   How long does it take to service a request

2. **Traffic**

   Measure of how much demand is being placed on your system

3. **Errors**

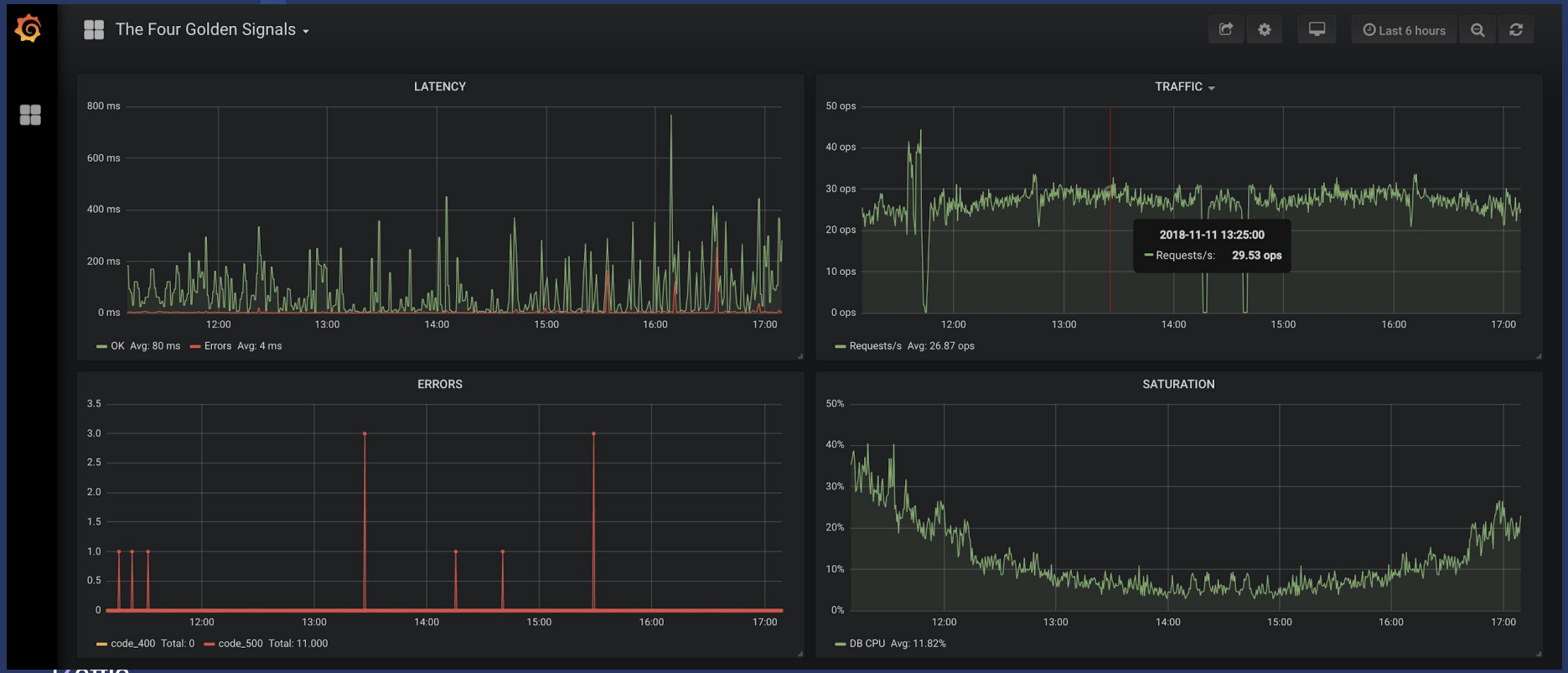
   The rate of requests that fail

4. **Saturation**

   How “full” your service is

iZettle

Dashboards for Starters



Log Querying for Deep Diving

8 hits

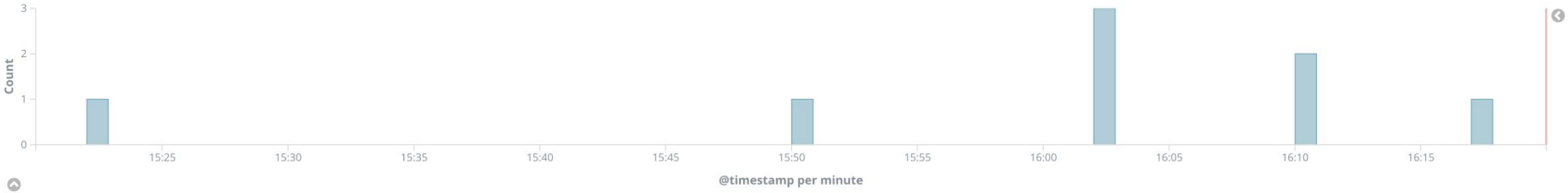
New Save Open Share Reporting Last 1 hour

kubernetes.labels.release:"legacy-api" AND method:GET AND NOT status:[200 TO 299]

Uses lucene query syntax

Add a filter +

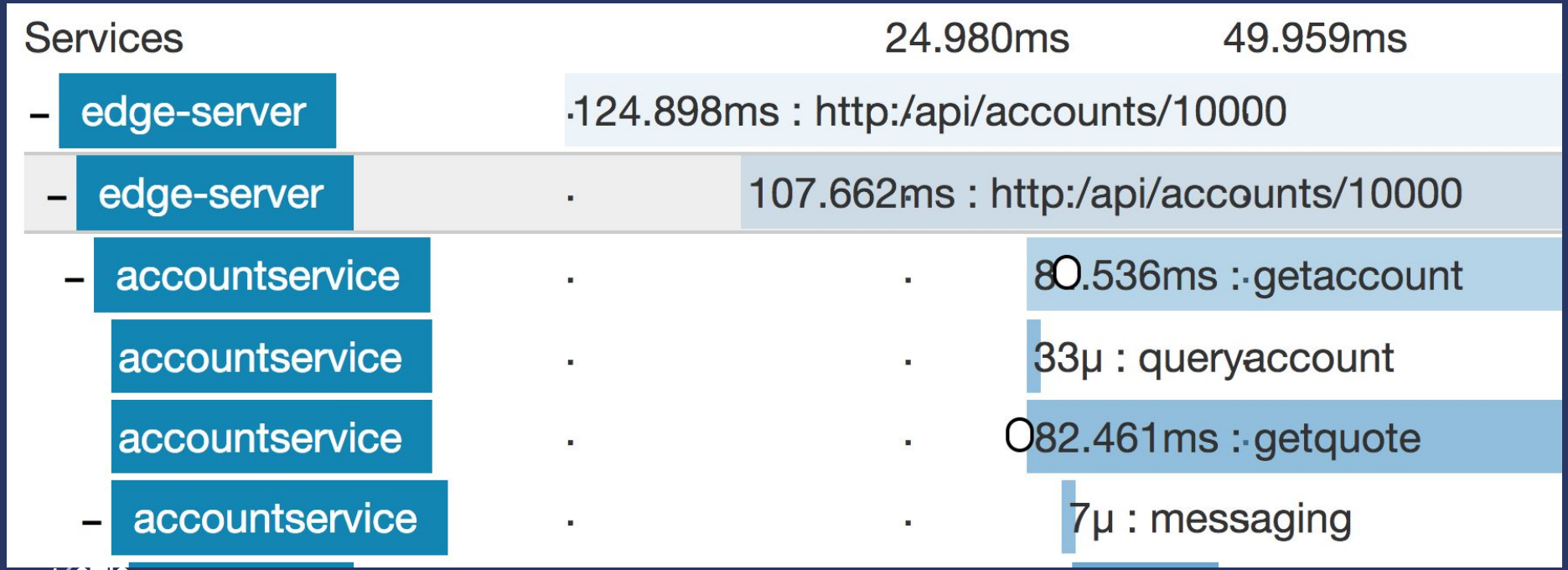
November 11th 2018, 15:19:58.355 - November 11th 2018, 16:19:58.355 — Auto



Time	_source
November 11th 2018, 16:17:40.000	method: GET kubernetes.labels.release: legacy-api time: November 11th 2018, 16:17:40.000 remoteIP: 100.97.195.64 host: legacy.iposws.com url: /v2/timesheets/userStatus/9939e2ad-d652-4c7b-a8b7-12b4a1f66358/self query: ? status: 403 userAgent: iZettlePro/3 CFNetwork/894 Darwin/17.4.0 referer: - request-id: - remote-app-version: 4.25.0 remote-app-build: 3 log: { "time":"2018-11-11T16:17:40", "remoteIP":"100.97.195.64", "host":"legacy.iposws.com", "url":"/v2/timesheets/userStatus/9939e2ad-d652-4c7b-a8b7-12b4a1f66358/self", "query":"?", "method":"GET", "status":403, "userAgent":"iZettlePro/3 CFNetwork/894 Darwin/17.4.0", "referer":"-", "request-id": "-", "remote-app-version": "4.25.0", "remote-app-build": "3" } stream: stdout docker.container_id: 34a226788c17493a2de17b3667e7e54e390efb6b409e134dca637f31a1d01e71 kubernetes.container_name: legacy-api kubernetes.namespace_name:
November 11th 2018, 16:10:21.000	method: GET kubernetes.labels.release: legacy-api time: November 11th 2018, 16:10:21.000 remoteIP: 100.106.237.64 host: 34.250.186.3 url: /index.php query: status: 404 userAgent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:47.0) Gecko/20100101 Firefox/47.0 referer: - request-id: - remote-app-version: - remote-app-build: - log: { "time":"2018-11-11T16:10:21", "remoteIP":"100.106.237.64", "host":"34.250.186.3", "url":"/index.php", "query":"", "method":"GET", "status":404, "userAgent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:47.0) Gecko/20100101 Firefox/47.0", "referer":"-", "request-id": "-", "remote-app-version": "-", "remote-app-build": "-" } stream: stdout docker.container_id: 4ebb9d68bc0baa4c2d84e8fa3f17c23e1eacf60ceb2e07042ab6dd6a2d28596d kubernetes.container_name: legacy-api kubernetes.namespace_name: default

iZettle

Distributed Tracing for Following Users Experience



iZettle

Incident Response

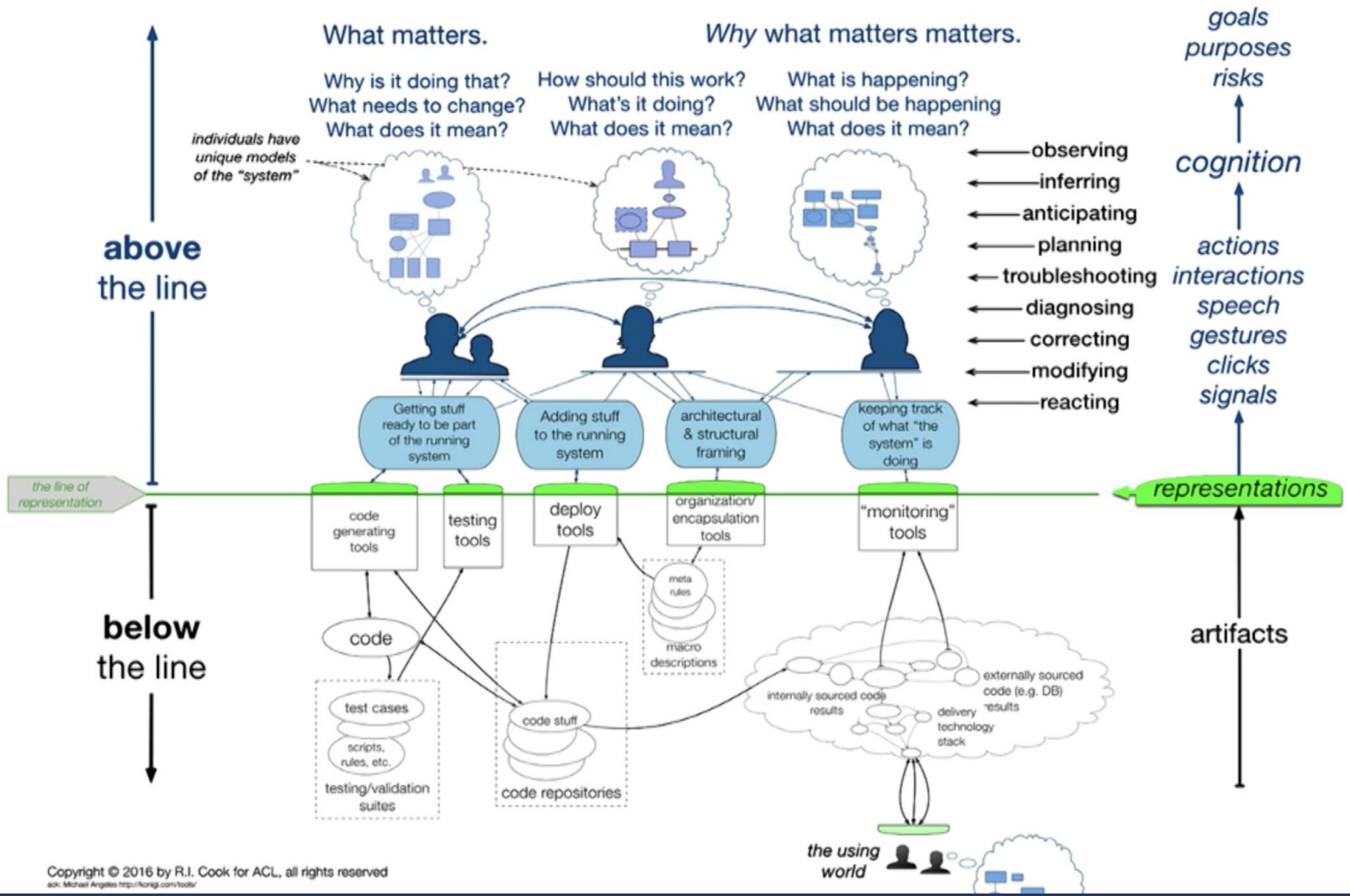
iZettle

You Can’t See Code

iZettle

Everyone has their own mental model of the system

iZettle



ack: Michael Angeles http://kongi.com/tools/

Track Outages

What was the Time to Detection?
What was the Time to Recovery?
How many users were affected?
How many times in a period did outage happen?

iZettle

Post Incident Reviews

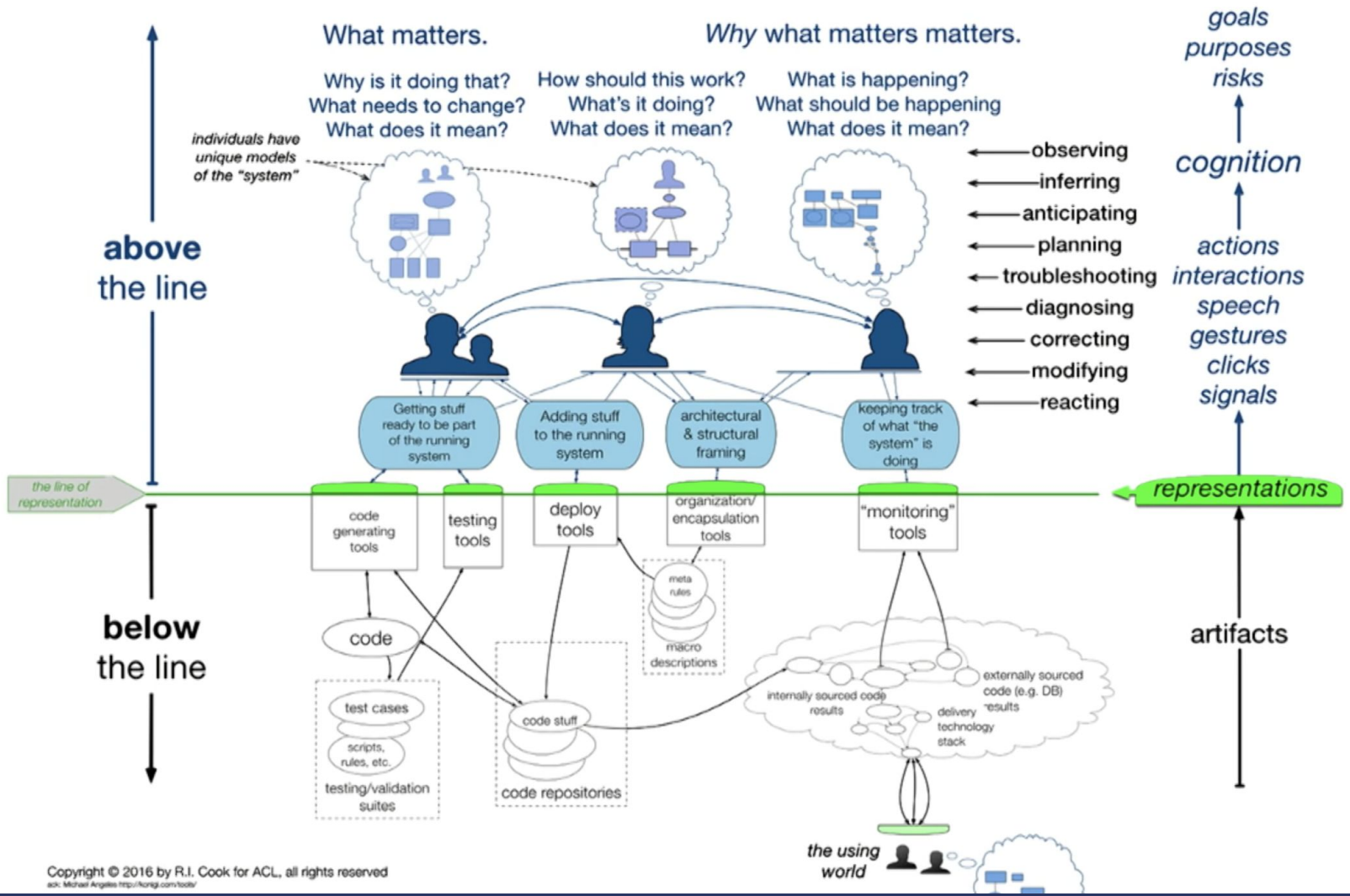
iZettle

Post Incident Review

- Ensuring the incident is documented
- All contributing root causes are understood
- Effective preventive actions are put in place to reduce recurrence

How different people share their mental model of the system

iZettle



ack: Michael Angeles http://kongi.com/tools/

SNAFU CATCHERS

STELLA

Report from the SNAFUcatchers Workshop on Coping With Complexity

Brooklyn NY, March 14-16, 2017

Winter storm STELLA

Woods' Theorem: *As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases rapidly.*

iZettle

Blameless PostMortems and a Just Culture

Posted by **John Allspaw** on May 22, 2012

Last week, Owen Thomas wrote a flattering article over at Business Insider on how we handle errors and mistakes at Etsy. I thought I might give some detail on how that actually happens, and why.

Anyone who's worked with technology at any scale is familiar with failure. Failure cares not about the architecture designs you slave over, the code you write and review, or the alerts and metrics you meticulously pore through.

So: failure happens. This is a foregone conclusion when working with complex systems. But what about those failures that have resulted due to the actions (or lack of action, in some cases) of individuals? What do you do with those careless humans who caused everyone to have a bad day?

…participants should be able to give a detailed account ***without fear of punishment or retribution.***

iZettle

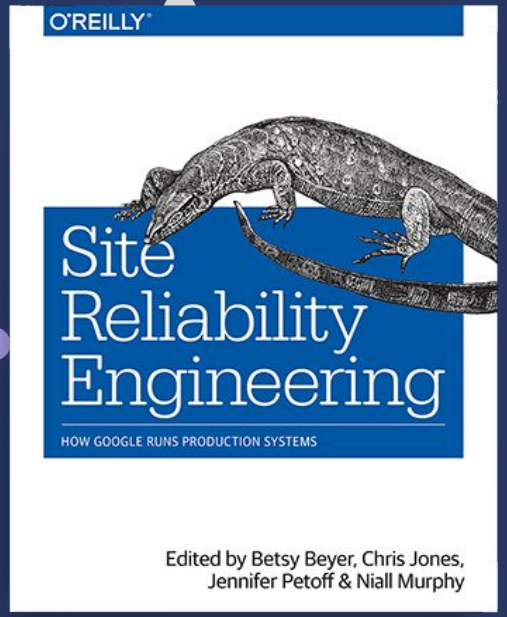
Share Reviews So Other Teams Can Learn From Your Incidents

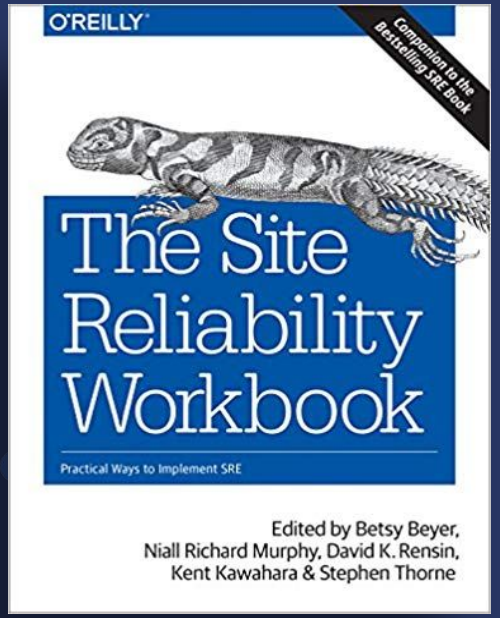
iZettle

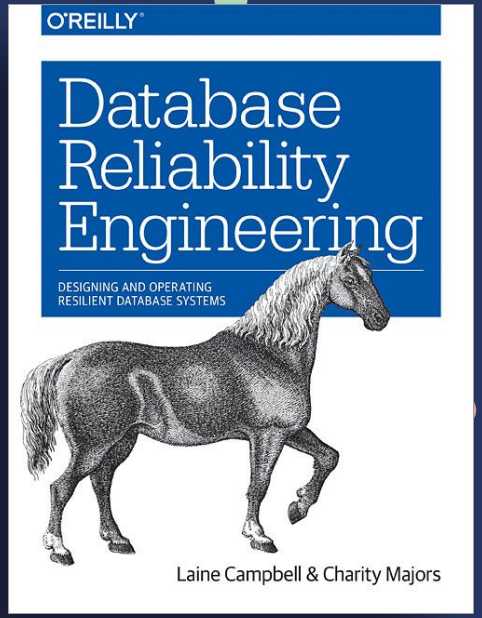
iZettle

Summary

1. Failure happens
2. How we manage failure is key
3. What can we do to make tomorrow better than today?


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy


O'REILLY
Companion to the Bestselling SRE Book
The Site Reliability Workbook
Practical Ways to Implement SRE
Edited by Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara & Stephen Thorne


O'REILLY
Database Reliability Engineering
DESIGNING AND OPERATING RESILIENT DATABASE SYSTEMS
Laine Campbell & Charity Majors

iZettle