A Layered Approach to Progressive Delivery

split

@SplitSoftware

Do you do this when your team releases to production?

Can you remember
a release night
that went like this?

https://imgflip.com/user/Knightrogens
https://imgflip.com/i/1mp8tq

How do you respond when someone asks, “How successful was that release?”

https://www.flickr.com/photos/tomhilton/16686579989

split

It doesn't have to be that way!

linkedin.com/in/davekarow

What I'm up to: demystifying progressive delivery, especially the role of automated data attribution, early in partial releases.

How I got here: Three decades of experience in developer tools, developer communities, and evangelizing sustainable software delivery practices that deliver impact, without burning out humans.

Where I've been: DHL Worldwide Express, Sun Microsystems, Gupta Technologies, Remedy Software, Marimba (BMC), Keynote Systems (Dynatrace), SOASTA (Akamai), BlazeMeter (CA/Broadcom) and now Split Software.

@SplitSoftware

A Layered Approach to Progressive Delivery

Build Your Way Up to Faster, Safer, Smarter Releases

01 Progressive Delivery: What Is It, Really?

02 Role Models: Progressive Delivery In The Wild

03 The Foundation: Decouple Deploy from Release

04 The Upper Layers: Data-Informed Practices, Automated

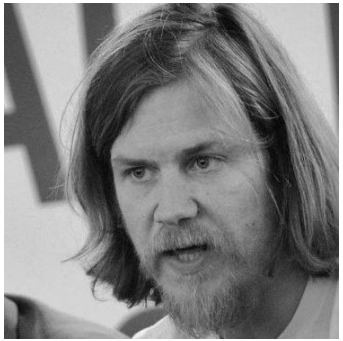
Progressive Delivery
What Is It, Really?

The Roots of “Progressive Delivery”

@SamGuckenheimer

“Well, when we’re rolling out services. What we do is ***progressive experimentation*** *because what really matters is the blast radius.* ***How many people will be affected when we roll that service out and what can we learn from them?”***

Sam Guckenheimer, quoted in
https://www.infoq.com/presentations/progressive-delivery/



@monkchips
(James Governor)

The Roots of “Progressive Delivery”

RedMonk

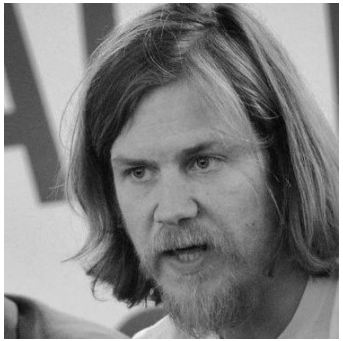
the developer-focused industry analyst firm

Videos Research Events About Team Services Clients

JAMES GOVERNOR'S MONKCHIPS

Towards Progressive Delivery

By James Governor | @monkchips | August 6, 2018



@monkchips

...a new basket of skills and technologies concerned with modern software development, testing and deployment.

Progressive Delivery is the next step after Continuous Delivery,
where new versions are deployed to a subset of users
and are evaluated in terms of correctness and performance
before rolling them to the totality of the users
and rolled back if not matching some key metrics.

Carlos Sanchez
(Sr. Cloud Software Engineer @ Adobe)

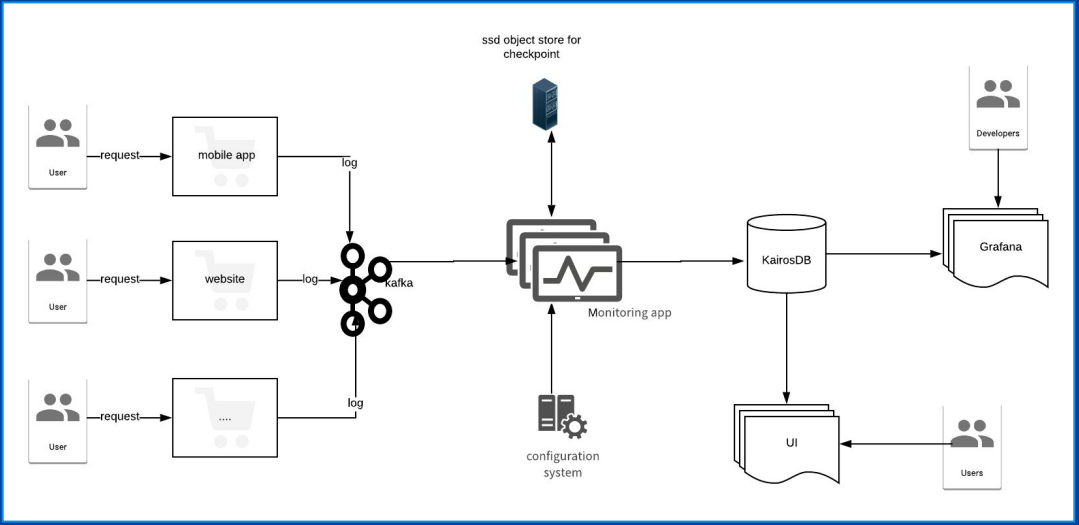
https://blog.csanchez.org/2019/01/22/progressive-delivery-in-kubernetes-blue-green-and-canary-deployments/

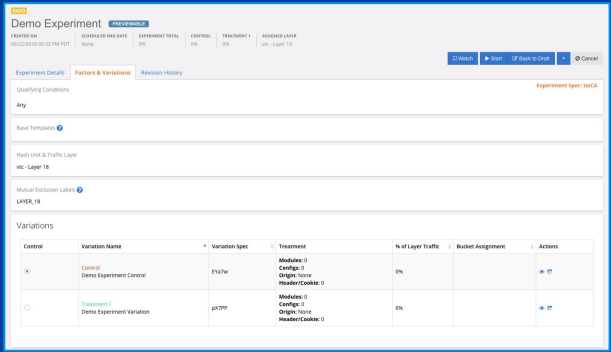
02

Role Models:
Progressive Delivery In the Wild

Example:
Walmart EXPO

Test to learn	Test to launch
• **Purpose:** Understand customer behavior and validate or invalidate a hypothesis • Feature likely only launches if it is better than the production experience • Ex. Guest Cart	• **Purpose:** Mitigating risk by phasing out the rollout to customers, and ensuring no bugs are introduced • Feature is necessary to launch for the business. Will launch if it is doing no harm • Ex. ADA

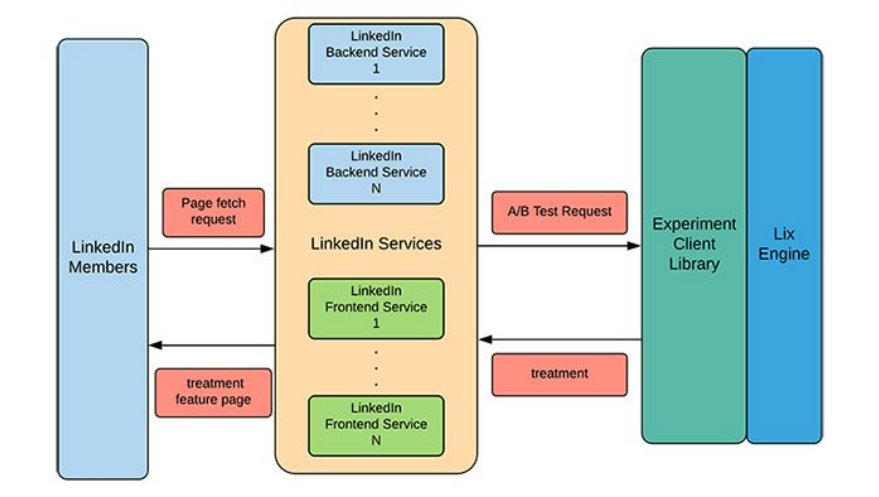


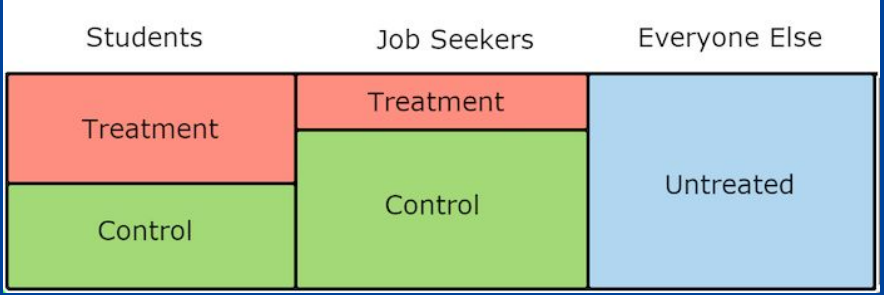


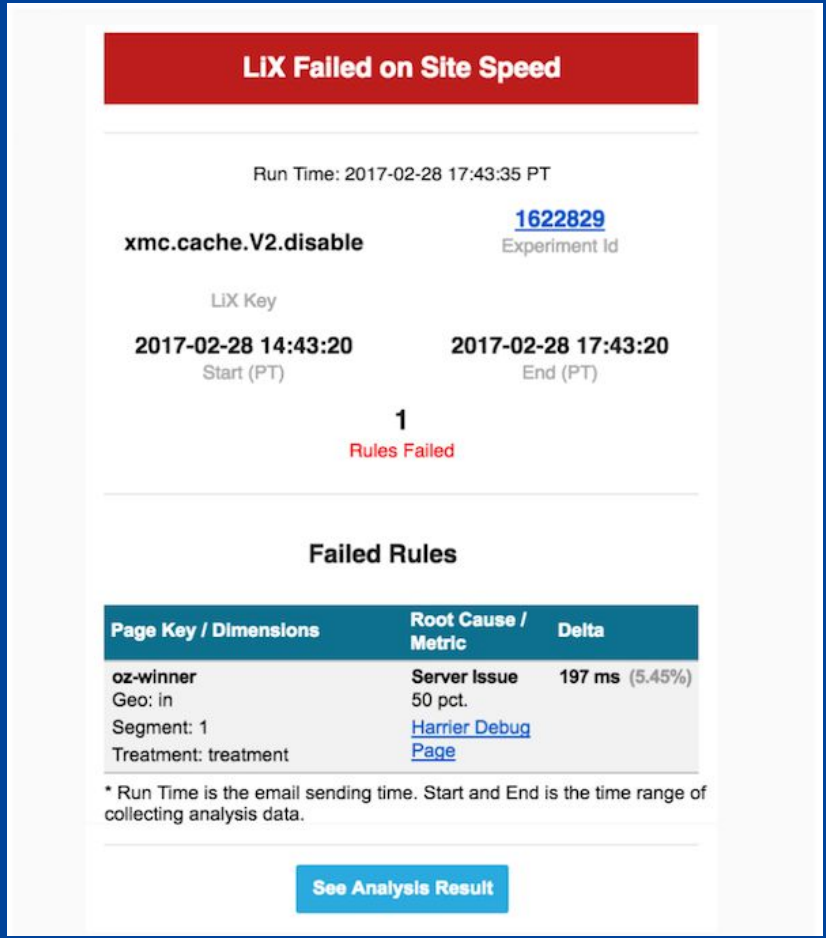
It's All About the Data

When someone runs an A/B test, they expect results, and whether they are positive or negative, they need to be correct. One challenge we faced in building our own platform was building trust in the results. With the support of a strong product analytics team, and a revamped data pipeline, we were able to achieve that.

Example:
LinkedIn LiX

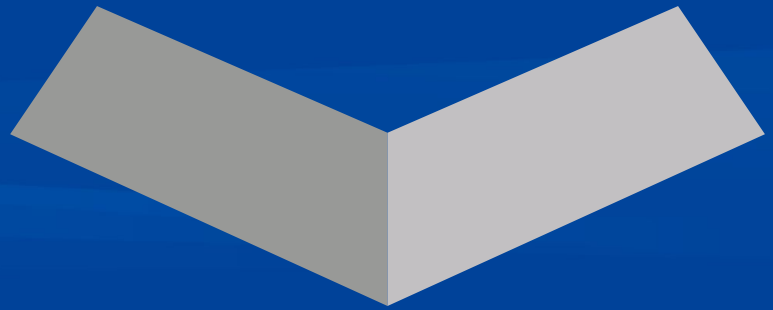






03

The Foundation:
Decouple Deploy from Release



4 Ways to Decouple Deploy From Release (How You Roll Matters)

Benefits \ Approach	Blue/Green Deployment	Canary Release (container based)	Feature Flags	Feature Flags + Data, Integrated
Avoid Downtime				
Limit The Blast Radius				
Limit WIP / Achieve Flow				
Learn During The Process				

https://www.split.io/blog/learn-the-four-shades-of-progressive-delivery/

split

@SplitSoftware

4 Ways to Decouple Deploy From Release (How You Roll Matters)

Benefits \ Approach	Blue/Green Deployment	Canary Release (container based)	Feature Flags	Feature Flags + Data, Integrated
Avoid Downtime				
Limit The Blast Radius				
Limit WIP / Achieve Flow				
Learn During The Process				

https://www.split.io/blog/learn-the-four-shades-of-progressive-delivery/

4 Ways to Decouple Deploy From Release (How You Roll Matters)

Benefits \ Approach	Blue/Green Deployment	Canary Release (container based)	Feature Flags	Feature Flags + Data, Integrated
Avoid Downtime				
Limit The Blast Radius				
Limit WIP / Achieve Flow				
Learn During The Process				

https://www.split.io/blog/learn-the-four-shades-of-progressive-delivery/

split

@SplitSoftware

4 Ways to Decouple Deploy From Release (How You Roll Matters)

Benefits \ Approach	Blue/Green Deployment	Canary Release (container based)	Feature Flags	Feature Flags + Data, Integrated
Avoid Downtime				
Limit The Blast Radius				
Limit WIP / Achieve Flow				
Learn During The Process				

https://www.split.io/blog/learn-the-four-shades-of-progressive-delivery/

split

@SplitSoftware

4 Ways to Decouple Deploy From Release (How You Roll Matters)

Benefits \ Approach	Blue/Green Deployment	Canary Release (container based)	Feature Flags	Feature Flags + Data, Integrated
Avoid Downtime				
Limit The Blast Radius				
Limit WIP / Achieve Flow				
Learn During The Process				

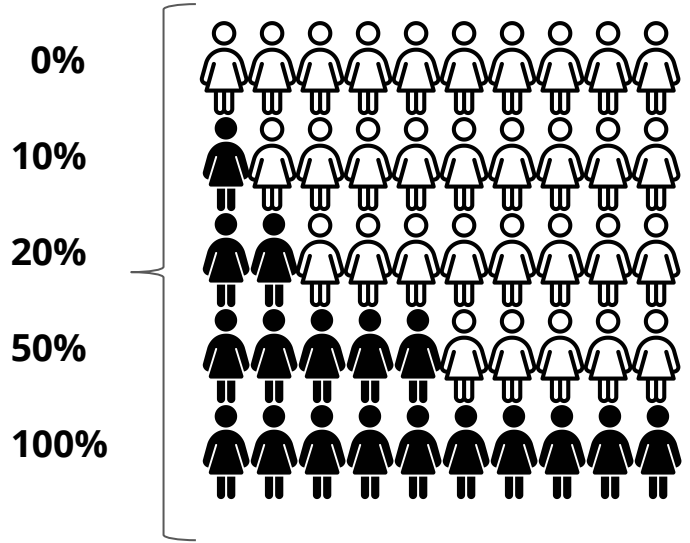
https://www.split.io/blog/learn-the-four-shades-of-progressive-delivery/

split

@SplitSoftware

Feature Flag

Like a dimmer switch for changes



What a Feature Flag Looks Like In Code

Simple “on/off” example

```
treatment = flags.getTreatment(“related-posts”);
if (treatment == “on”) {
    // show related posts
} else {
    // skip it
}
```

What a Feature Flag Looks Like In Code

Multivariate example

```
treatment = flags.getTreatment("search-algorithm");
if (treatment == "v1") {
    // use v1 of new search algorithm
} else if (feature == "v2") {
    // use v2 of new search algorithm
} else {
    // use existing search algorithm
}
```

Foundational Capabilities Unlocked By Feature Flags

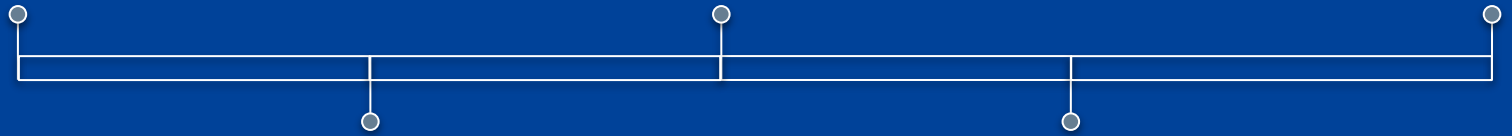
- Incremental Feature Development for Flow
- Testing In Production
- Kill Switch (big red button)

04

The Upper Layers:
Data-Informed Practices, Automated

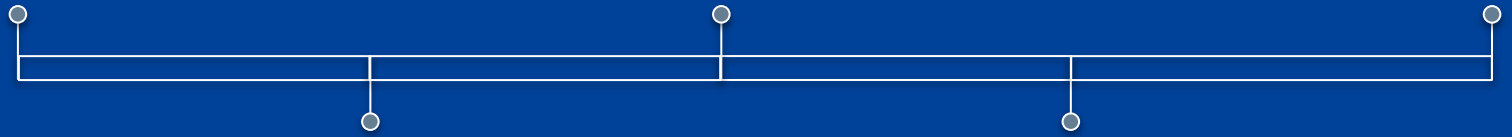
Why Automate Data-Informed Practices?
A Different Way to “Ship” Becomes Possible

DEPLOY
Code deployed
No exposure



Why Automate Data-Informed Practices?
A Different Way to "Ship" Becomes Possible

DEPLOY
Code deployed
No exposure

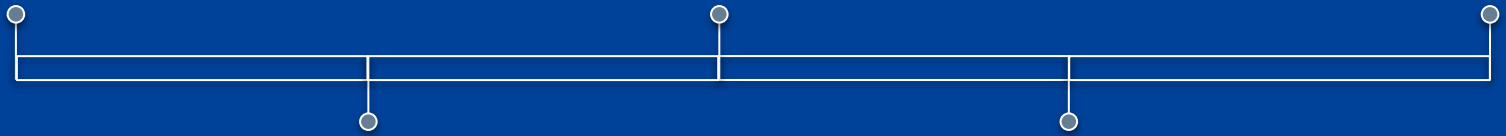


ERROR MITIGATION
0-50% Ramp
Identify bugs/crashes

Why Automate Data-Informed Practices?
A Different Way to "Ship" Becomes Possible

DEPLOY
Code deployed
No exposure

MEASURE
Maximum Power Ramp
Understand impact



ERROR MITIGATION
0-50% Ramp
Identify bugs/crashes

Why Automate Data-Informed Practices?
A Different Way to "Ship" Becomes Possible

DEPLOY
Code deployed
No exposure

MEASURE
Maximum Power Ramp
Understand impact

ERROR MITIGATION
0-50% Ramp
Identify bugs/crashes

SCALE MITIGATION
50-100% Ramp
Identify scaling issues

Why Automate Data-Informed Practices?
A Different Way to "Ship" Becomes Possible

DEPLOY
Code deployed
No exposure

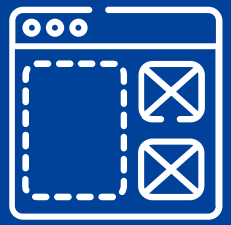
MEASURE
Maximum Power Ramp
Understand impact

RELEASE
Complete rollout

ERROR MITIGATION
0-50% Ramp
Identify bugs/crashes

SCALE MITIGATION
50-100% Ramp
Identify scaling issues

Heads Up! Not Everything Is As It Seems

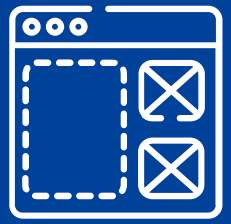


New Release

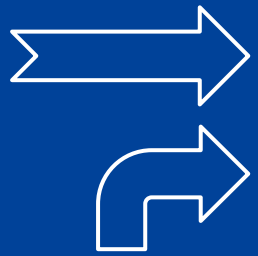
Metrics Change

“Can’t we just change things and monitor what happens?”

Problem To Solve: Separating Signal From Noise



New Release



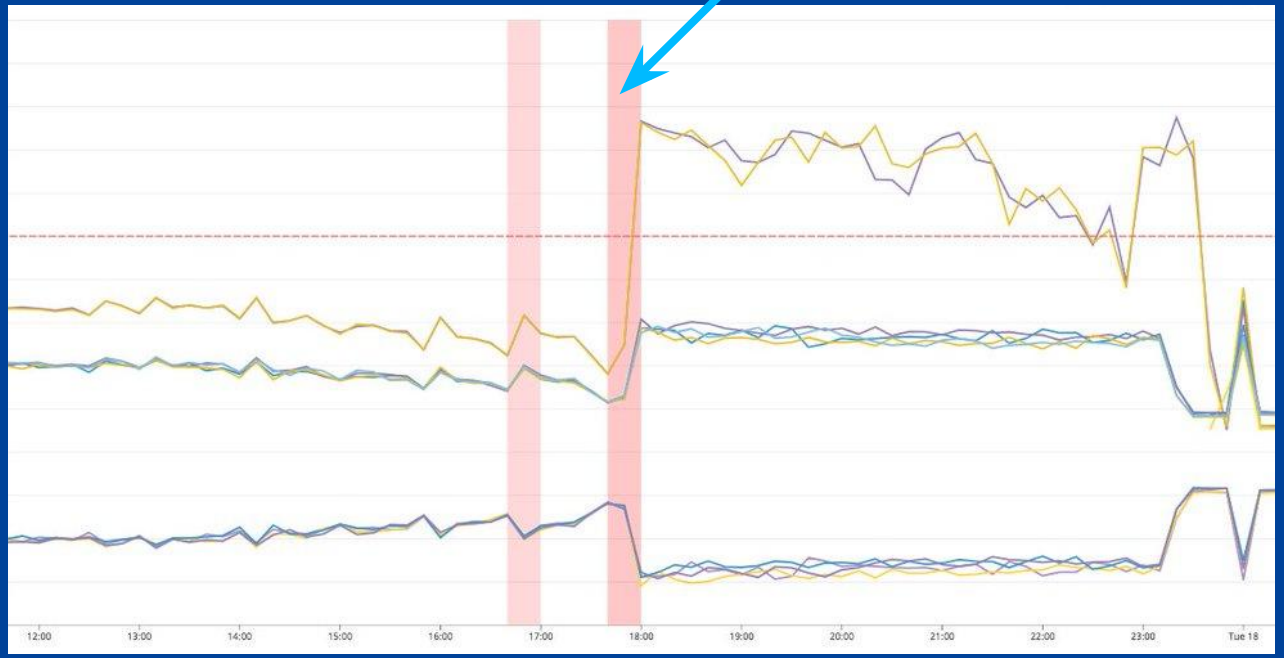
Everything else in the world

Metrics Change

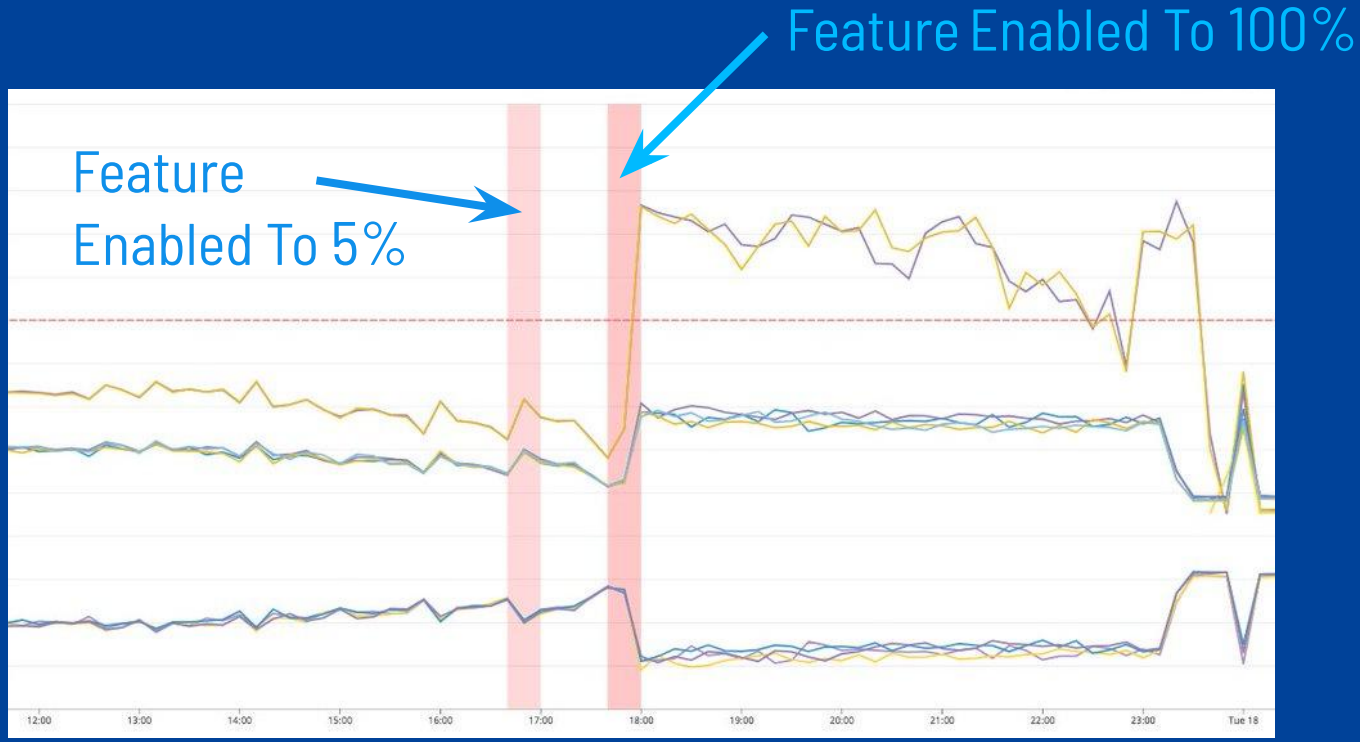
- Product changes
- Marketing campaigns
- Global Pandemics
- Nice Weather

Problem to Solve: Seeing Early Signs of Trouble

Feature Enabled To 100%

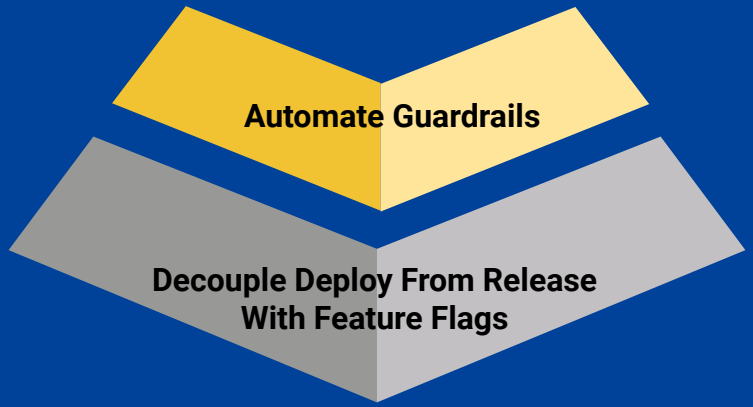


Problem to Solve: Seeing Early Signs of Trouble



Solution: Cancel Out External Influence With a Stats Engine
(Think noise cancelling headphones, but for your metrics)

WALK: Automate Guardrails/Do-No-Harm Metrics



- Alert on Exception / Performance Early In Rollout
- “Limit The Blast Radius” w/o Manual Heroics

Guardrail Metrics

Tasks Created per User

Significant **desired** impact

11.20%

Impact lies between
+8.71% and +14.74%

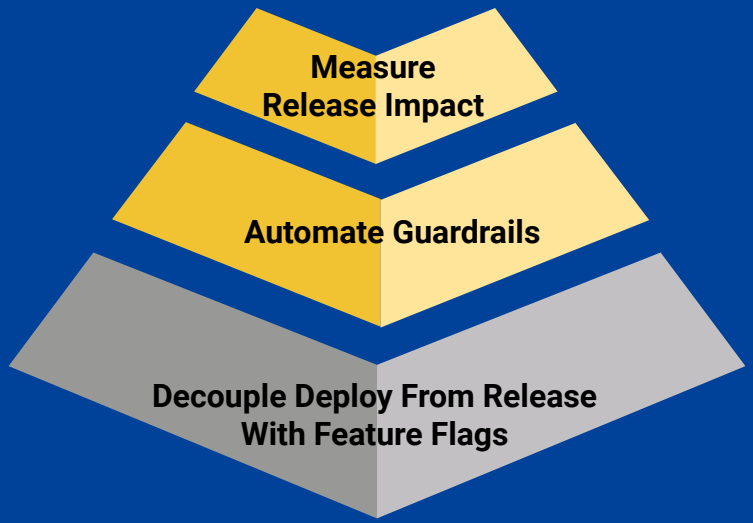
Initial Page Load Time

Significant **undesired** impact

25.64%

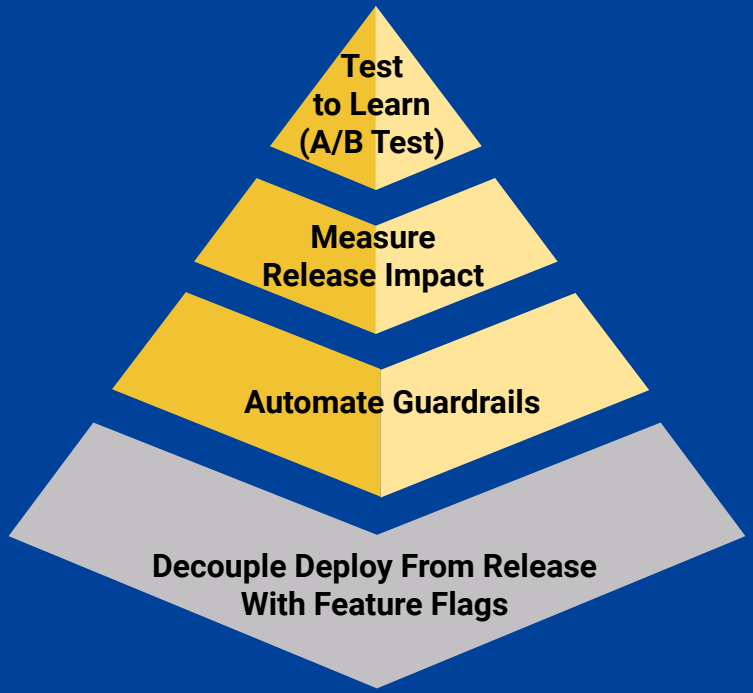
Impact lies between
-26.04% and -22.89%

RUN: Measure Release Impact



- Iteration w/o Measurement = Feature Factory 😡
- Direct Evidence of Our Efforts → Pride 😎

FLY: Test to Learn (A/B Test)



- Take Bigger Risks, Safely
- Learn Faster With Less Investment
  - Dynamic Config
  - Painted Door

This is What Sustainable Software Delivery Looks Like:

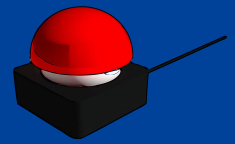
DEPLOY
Code deployed
No exposure

MEASURE
Maximum Power Ramp
Understand impact

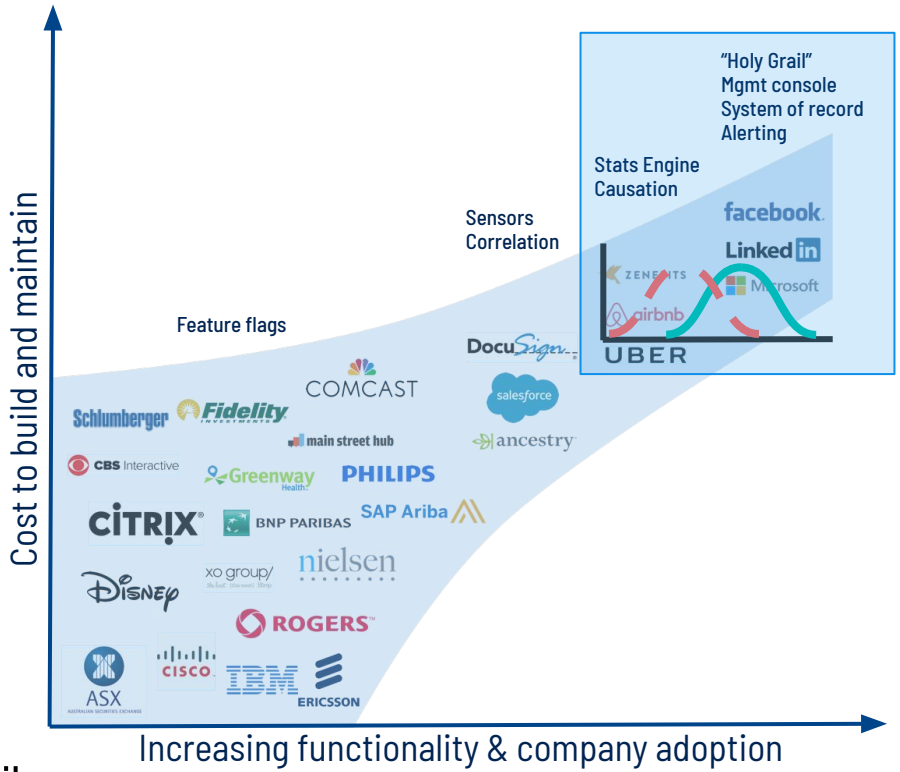
RELEASE
Complete rollout

ERROR MITIGATION
0-50% Ramp
Identify bugs/crashes

SCALE MITIGATION
50-100% Ramp
Identify scaling issues



How In-House Progressive Delivery Platforms Paved The Way For Split (They Proved the value of Layering Up, But It Required Big Investments)



Microsoft | > $50M annual cost

LinkedIn | > $30M annual cost

UBER | > $25M annual cost

Q&A + Booth Invitation

After the talk, come join me in the Split booth in the Expo Hall.

We're having fun with Confessions and Redemptions in Continuous Delivery.

We're also holding a raffle to win an Oculus Quest 2!

split