

# Building a DevOps culture

brian a. randell

partner, mcw technologies



Slides and links?

Slides and links

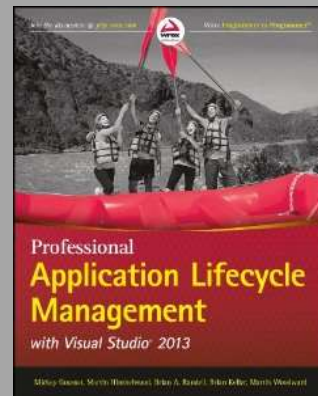
[\*\*http://bri.gd/brsa14content\*\*](http://bri.gd/brsa14content)

(will take you to my GitHub repo)

**Who am I?**

# Brian A. Randell

- Partner with MCW Technologies
- Microsoft MVP for Visual Studio ALM
- Member of Pluralsight's Technical Staff
- Co-author Pro ALM 2013 from Wrox
  - <http://bri.gd/bcazba02>



brianr@mcwtech.com | @brianrandell | blog.brianrandell.com



**Who are you?**

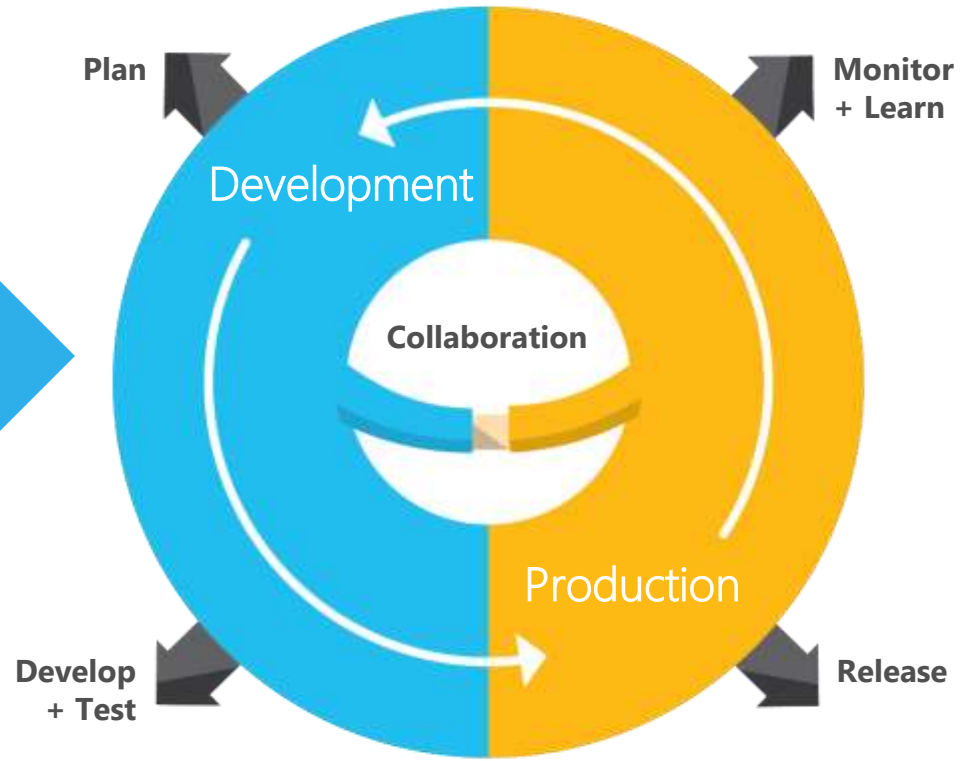


Every company is also  
a software company!





Today, modern software development is all about agility and time to market.



DevOps is an emerging trend that promises to increase the flow of value to customers.

“It has been said that the only sustainable advantage in business is the ability for a company to learn faster and respond more effectively than its competitors (also known as business agility).”

– CIO magazine

# What's DevOps?

DevOps is a full  
lifecycle investment



DevOps is a  
team undertaking



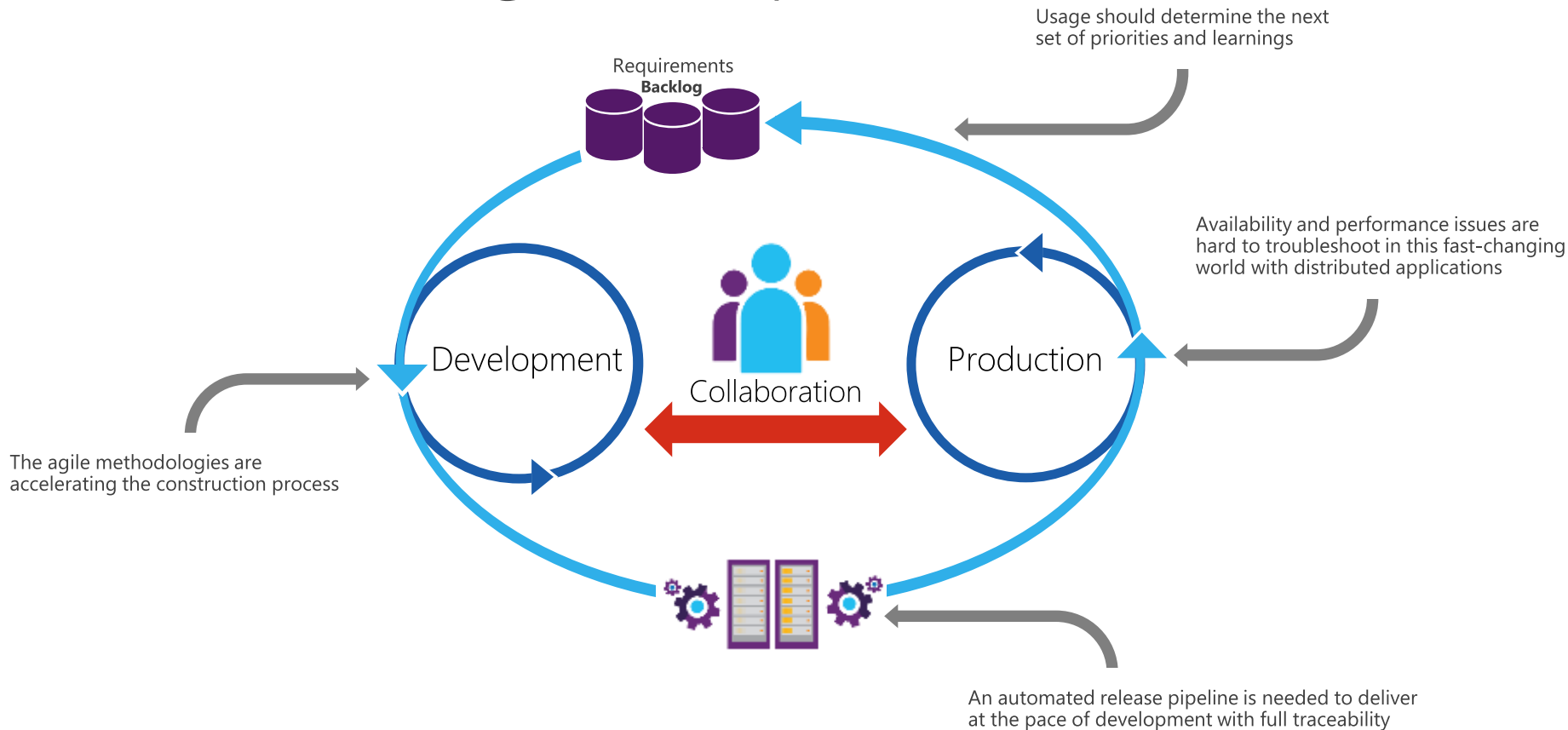
DevOps enables better  
software development  
and delivery practices



DevOps accelerates  
the last mile of  
continuous delivery



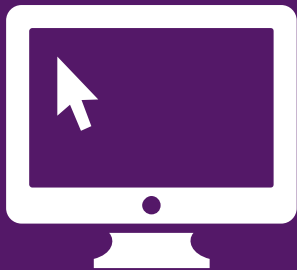
# What's driving DevOps?



# DevOps flavors

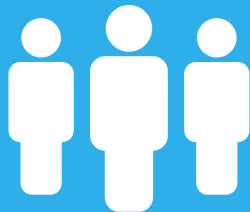
## WebOps

Google, Amazon, Twitter,  
Facebook, Xbox Live, etc.



## NoOps

Small web teams,  
start-ups



## Enterprise DevOps

Cross-functional  
organizations



# DevOps goals and success metrics

Increase  
deployment  
frequency

Reduce change  
lead-time (react  
faster to dynamic  
business needs)

Reduce change  
fail rate

Reduce mean-  
time-to-detect  
& repair (MTTD,  
MTTR)

**Agility** performance indicators

**Reliability** performance indicators

10+ deploys per day

# 10+ deploys a day

- Automated Infrastructure
- Shared Version Control
- One Step Build and Deploy
- Feature Flags / Branches
- Shared Metrics
- Communication

<http://bri.gd/xe10dday>



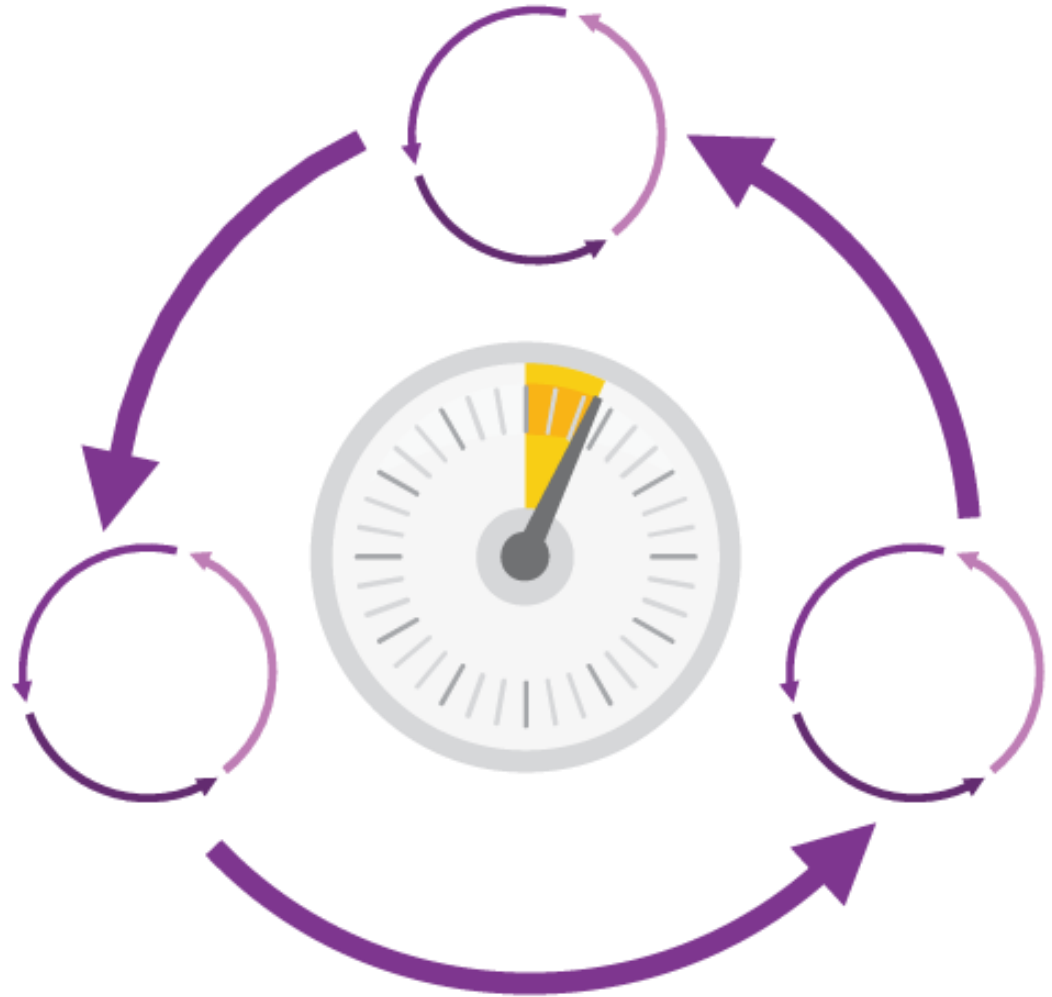
People

# Can't we all get along?

- Break down barriers
- Break down organizations
- Yes, start with a virtual team
- Move to "real" DevOps teams

# Interactions

Shorten  
cycle times



# Shorten cycle times

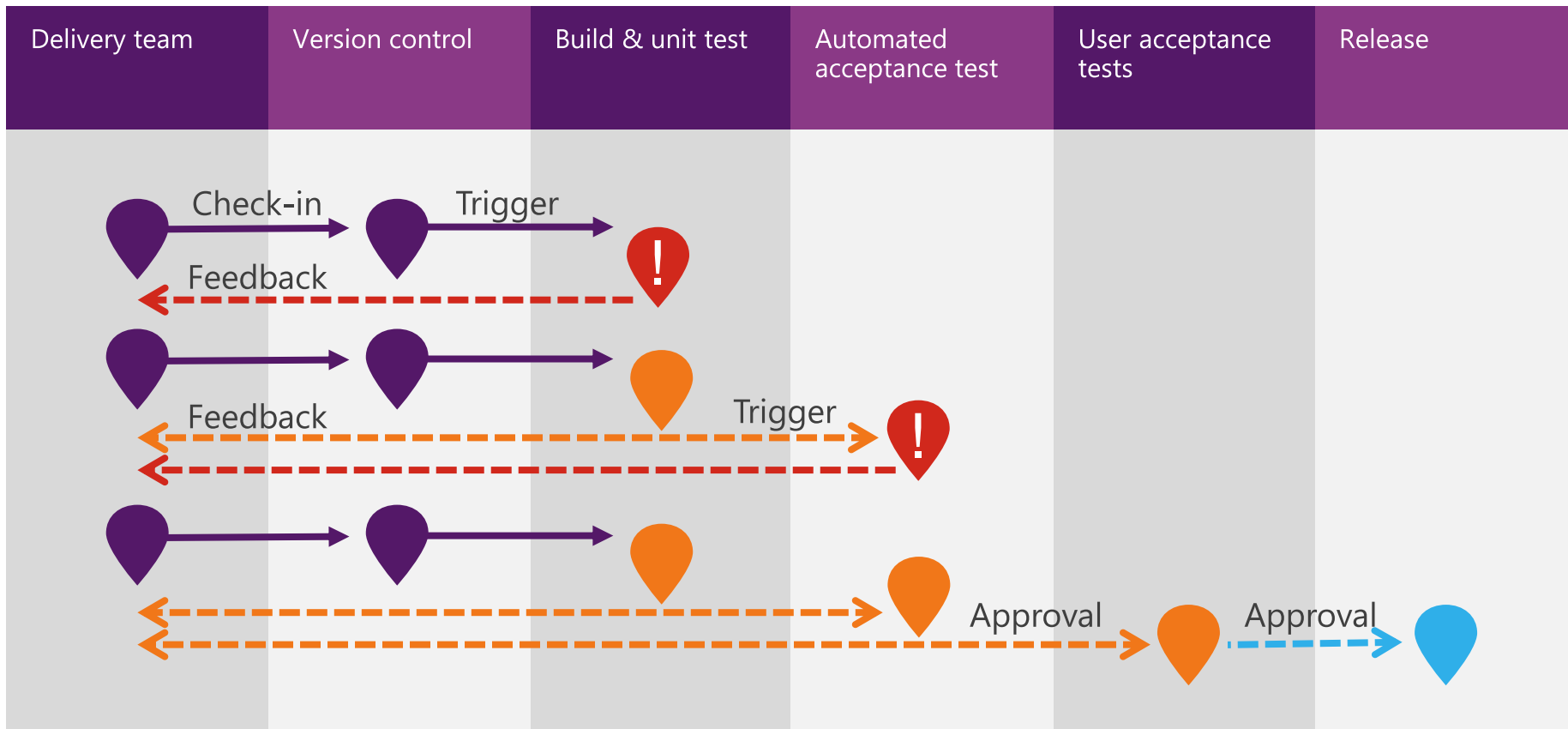
## Continuous delivery

Each release has a large "Release Tax" to pay

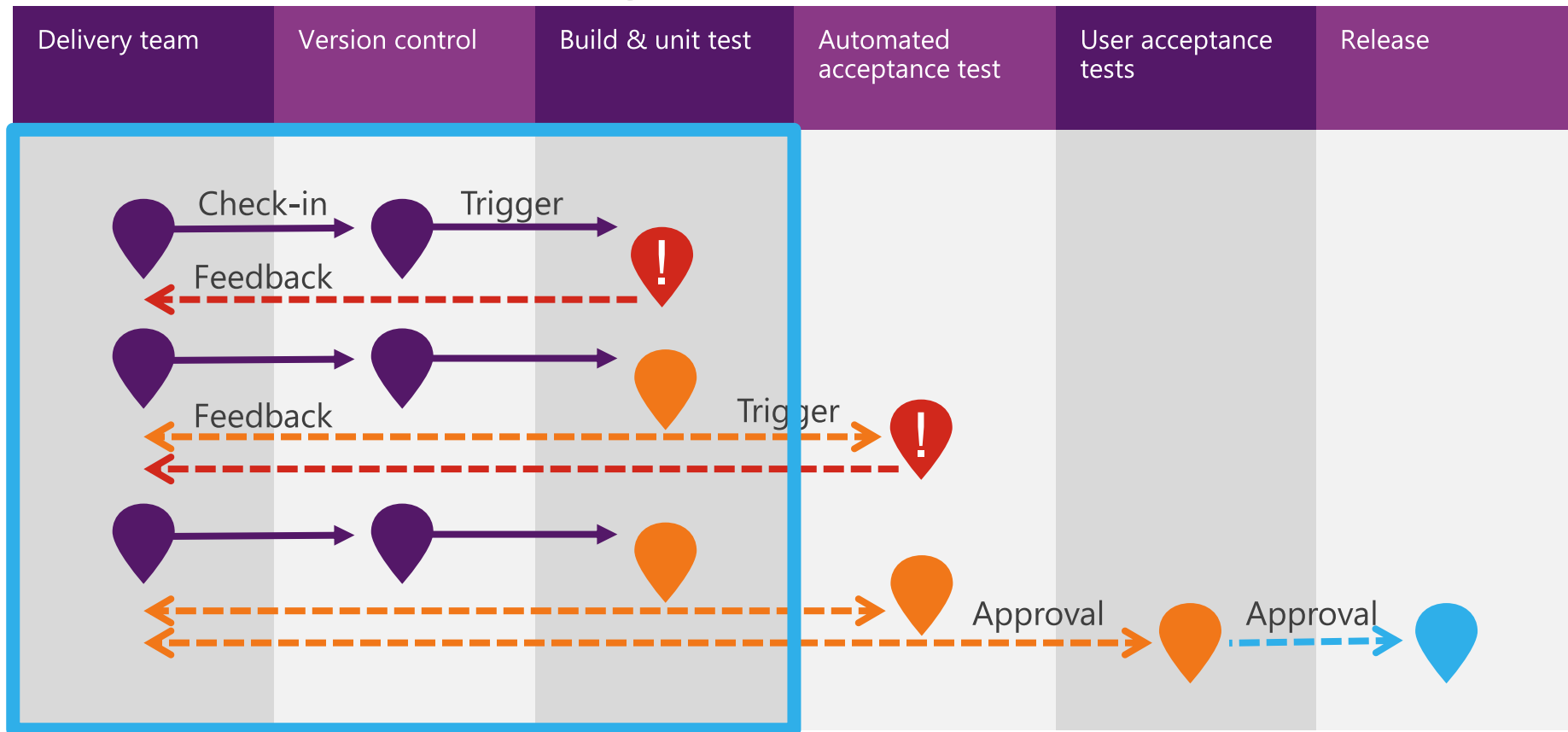
Developers work in agile iterations accumulating shippable bits many times before a release window is available

Manual testing is a significant effort  
Releases are high-risk requiring significant preparation

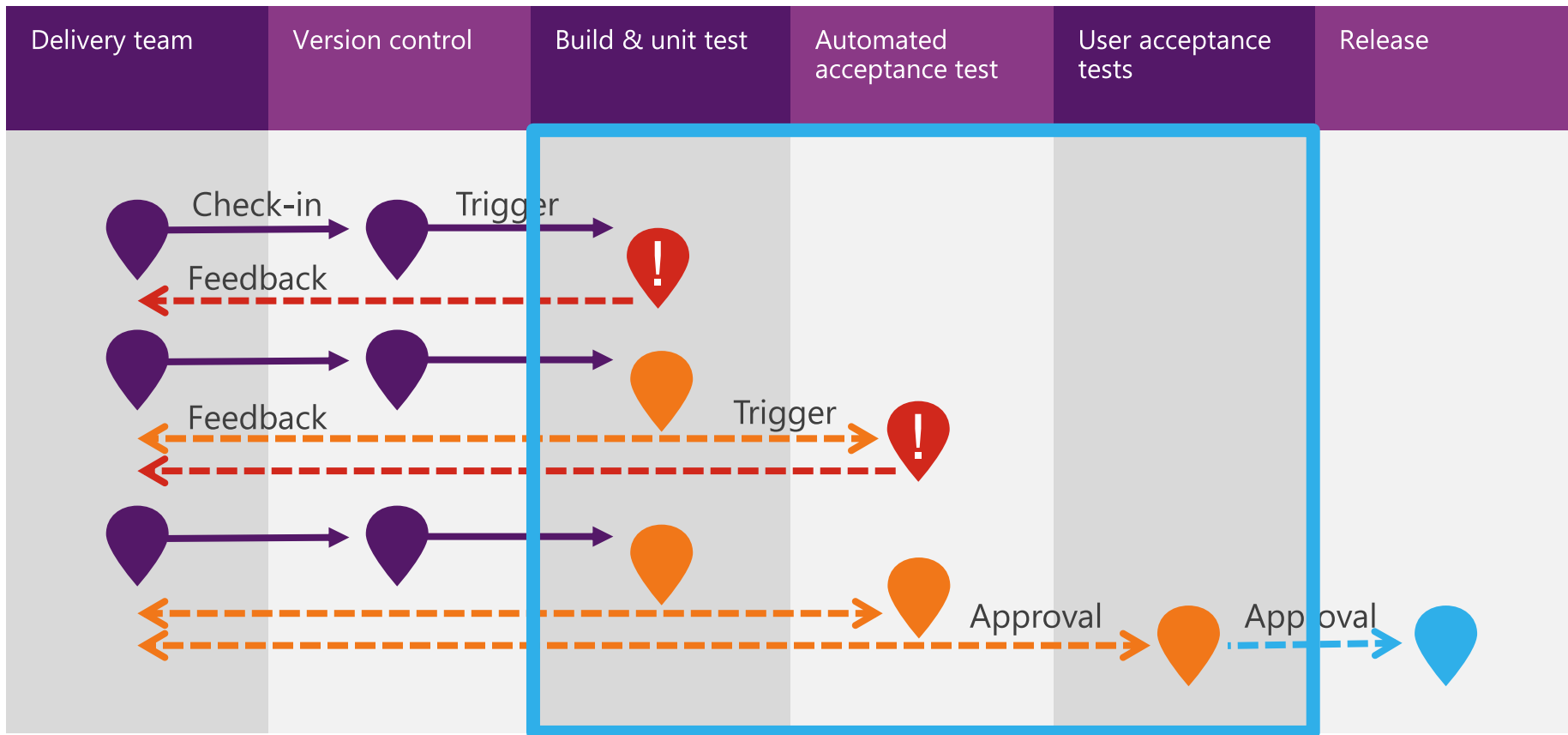
# Continuous delivery process



# Continuous integration

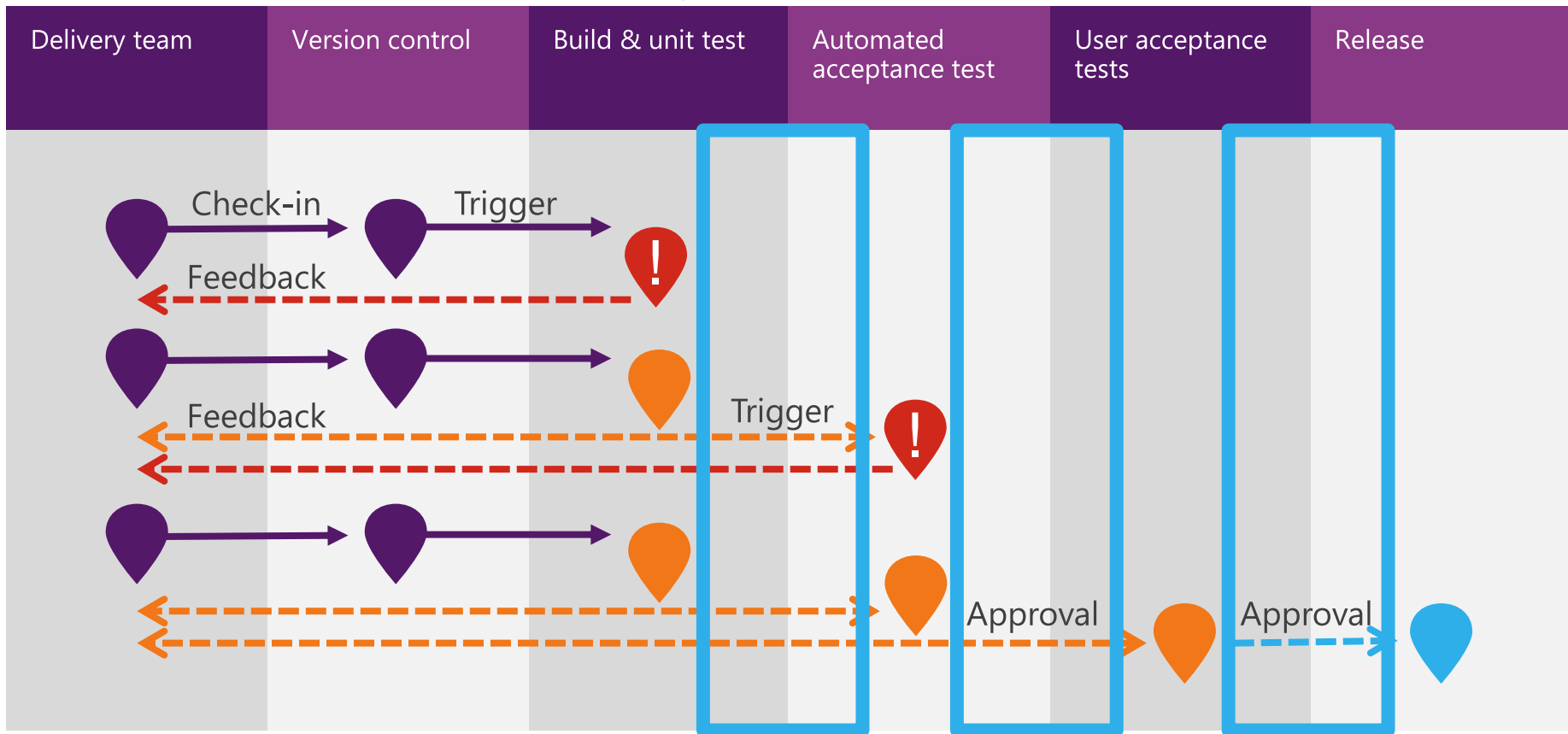


# Continuous inspection





# Continuous deployment



Improve quality  
& availability



# Improve quality & availability

Reducing defects that make it into production

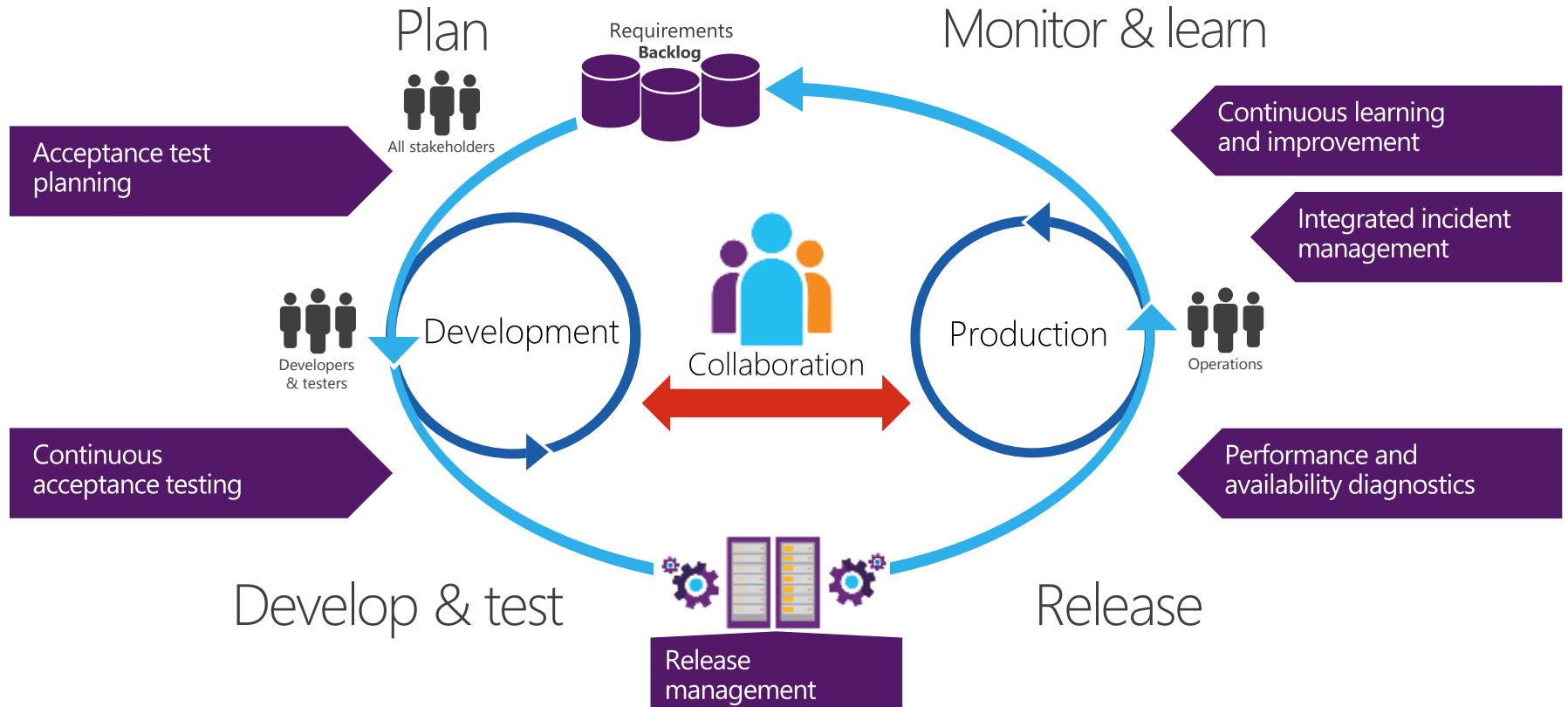
Defects arrive in production too frequently that ultimately impact your customers

A defect is simply a mistake that hasn't been discovered before production

Once the defect is in production, it's very difficult to detect and remediate

Need to drive down MTTD and MTTR  
Defects = poor customer experience  
= lost business

# Improve quality & availability



# Continuous learning

## Problems

Lack of information to  
prioritize future  
investments



## Solutions

Access to customer usage  
data to help you make  
informed decisions



## Value

Continuous learning



# DevOps benefits

Efficiency



Lower cost



Higher quality



Continuous learning



Increased flow of value

# Problem Areas

# Versioning Your REST Services

- URL versioning
  - You add a version number into the URL
  - `https://myendpoint.com/api/v2/app/resource`
- Custom request header
  - Same URL, but add some header like “api-vers
- Accept header with content type
  - “Accept: application/vnd.myapp.v2+json”

troyhunt.com

Observations, musings and conjecture about the world of software and technology

Your API versioning is wrong, which is why I decided ways

Monday, 10 February 2014

In the end, I decided the fairest, most balanced way was to piss everyone off equally. Of course, the great “tabs versus spaces” debate have I seen so many strong beliefs in entirely different

Imagine this:

HTTP GET:  
`https://haveIbeenpwned.com/api/breachedaccount/foe`

response:  
[“doobe”, “darker”]

This was just fine. When I built *Have I been pwned?* (HIBP) in late November, it was intended to be used. I think it's fair to say that the first two points were achieved, but not the last one. In week it was *more than Google Analytics could handle*. The point is that you can't always predict some point you may well need to change something that people already depend on.

Here's the problem – every time you start talking about anything to do with APIs over HT





# What about databases?

- You need rigor in your change management
  - “Data Architects” and DBAs need to be in full sync with “Service Architects” and developers
- You need to consider data migration as well as schema migration
- You might need to modernize your data models for the 21st century

# Tools

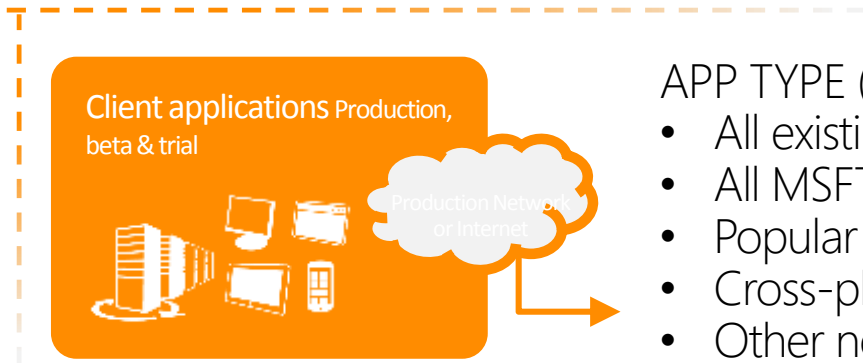
# Tooling

- Version Control
- Work Tracking
- Assembling and Packaging
- Deployment
- Monitoring
- Analytics

An example

# PreEmptive Analytics

# PreEmptive Analytics



## APP TYPE (Runtime/Device support)

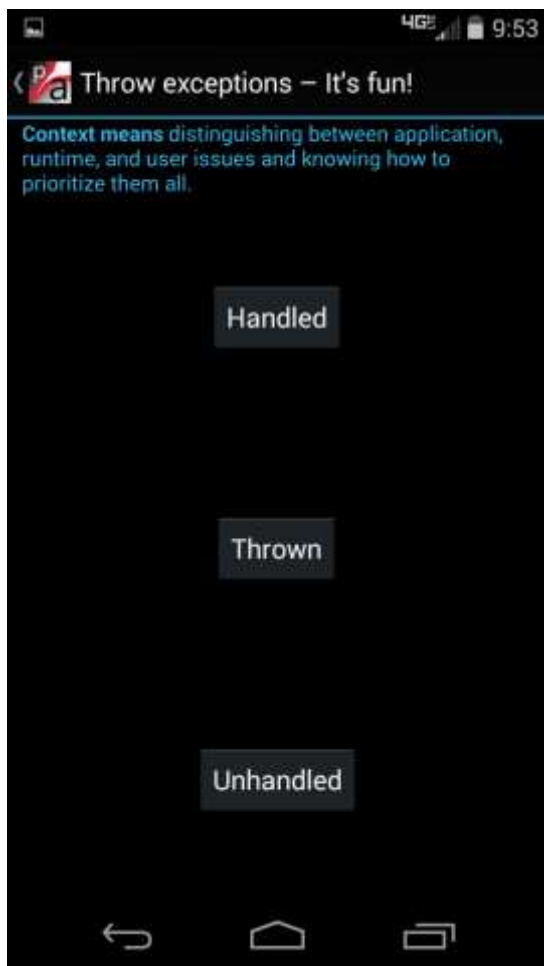
- All existing .NET, C++, and JavaScript Apps
- All MSFT modern surfaces (WinRT, WP, & JavaScript)
- Popular mobile options (iOS & Android)
- Cross-platform dev tools (Xamarin)
- Other non-MSFT runtimes (Java)

## USER TYPE (Application "Domain")

- Enterprise LoB
- Business-to-Business and Business-to-Consumer
- Commercial ISV

## DEPLOYMENT TYPE (Network infrastructure)

- Public (Internet), Private (corporate), Isolated (remote)





Overview



Critical Activity



Exceptions



Version Adoption



Target Utilization



Custom Data



Time and Location



Query Data

9/9/14 to 10/8/14 All Applications Refresh

Print Share

## Customer Activity

Customer	Total Sessions	Thrown Count	Unhandled Count	Caught Count	Unique Users
ABC Security	49	3	0	2	5
Cardoso Medical	19	8	0	2	0
Up And Away Inc.	20	18	2	4	0
Wro Garage	9	2	0	1	4
Solutions Inc.	12	1	0	4	6

## Happiness by Day

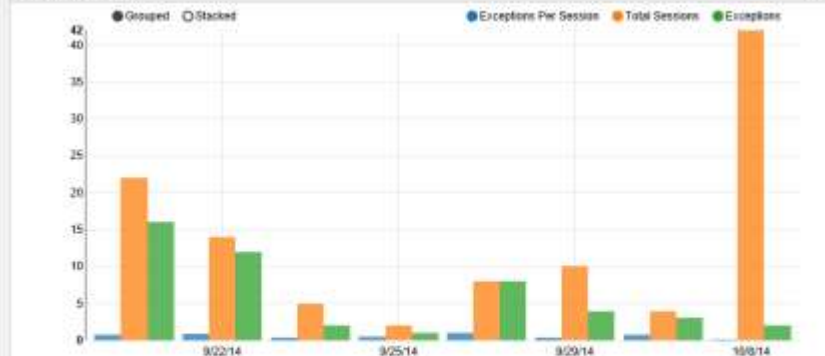


## Session Locations

## Activity by Role

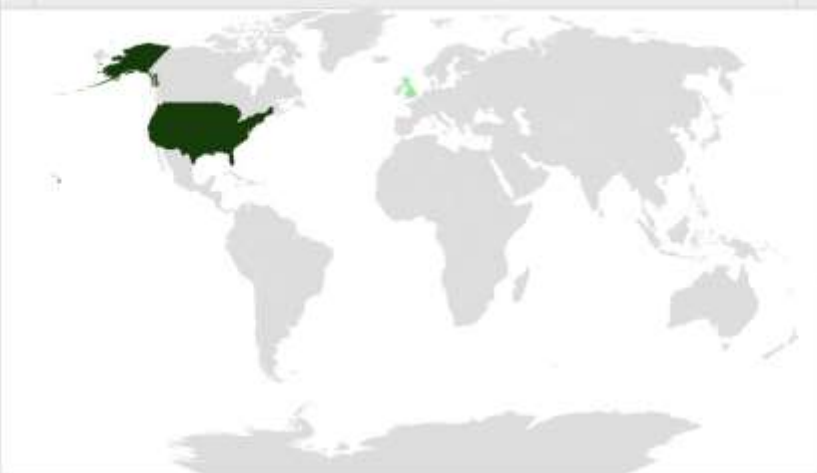
Department	Total Sessions	Thrown Count	Unhandled Count	Caught Count	Unique Users
Support	6	2	0	1	3
VP	10	7	0	2	3
Finance	44	10	0	4	4
Marketing	21	4	1	3	0
Evolution	0	3	4	0	2
Executive Management	2	0	0	0	2
Sales	12	2	2	3	4
IT Operations	2	0	0	0	2
Human Resources	6	0	0	0	5

## Service Level

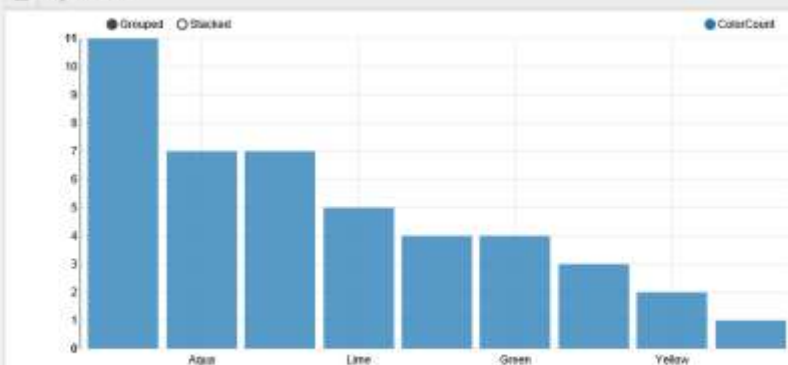


## Top 10 Colors

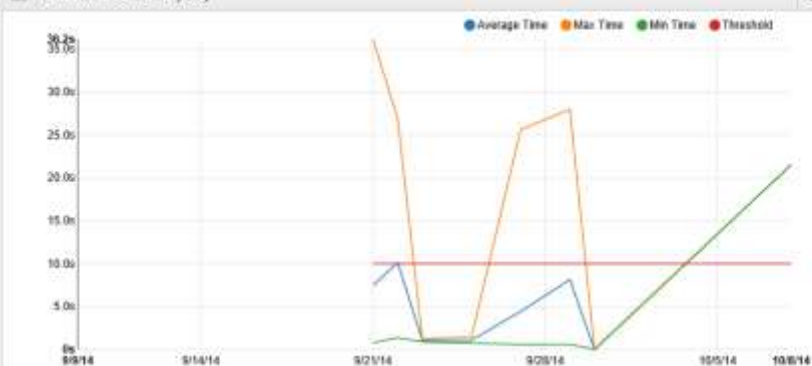
Session Locations



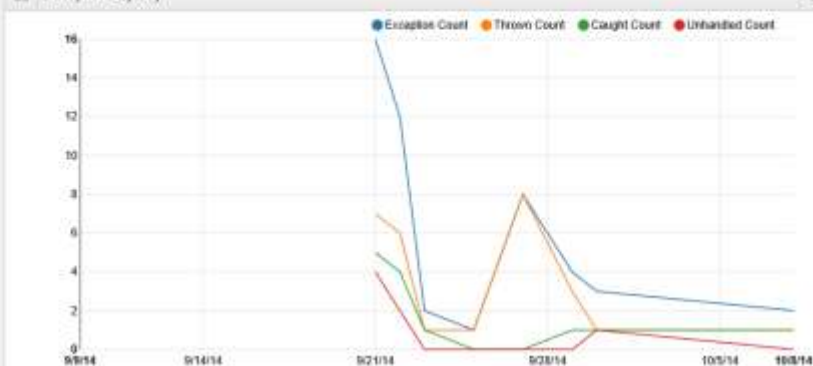
Top 10 Colors



Performance Times by Day



Exceptions by Day





Slides and links?

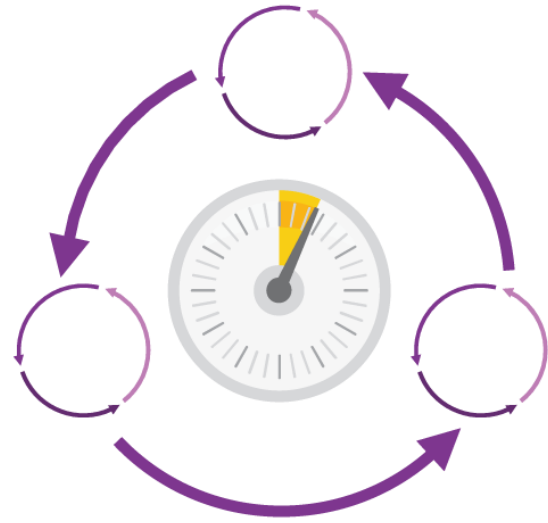
Slides and links

[\*\*http://bri.gd/brsa14content\*\*](http://bri.gd/brsa14content)

(will take you to my GitHub repo)

# Final thoughts

- DevOps is about ...
  - People
  - Interactions
  - Tools
- You don't buy DevOps
- Continue to evolve



thank you

uilding a  
DevOps cu

contact me

brian a. randell

partner, mcw technologies

[brianr@mcwtech.com](mailto:brianr@mcwtech.com)

[@brianrandell](#)

[blog.brianrandell.com](http://blog.brianrandell.com)

# Credits

- Images on slide 7
  - All from Office Online: search angry
- Content on slides 8-14, 21-30; graphic on 20 & 43
  - Matt Nunn, Microsoft
- Thanks to Preemptive Solutions for the demo app
  - <http://www.preemptive.com/>