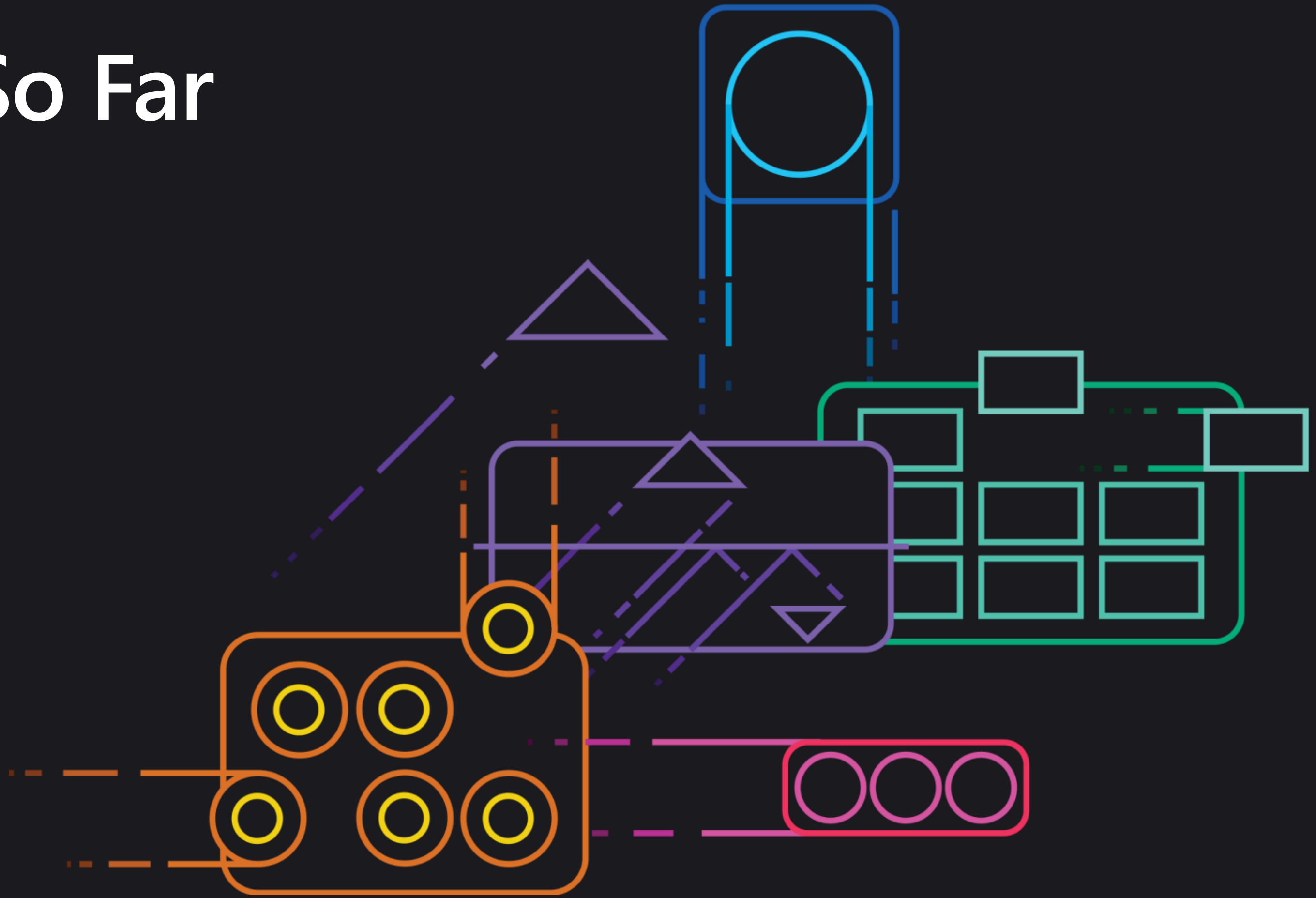
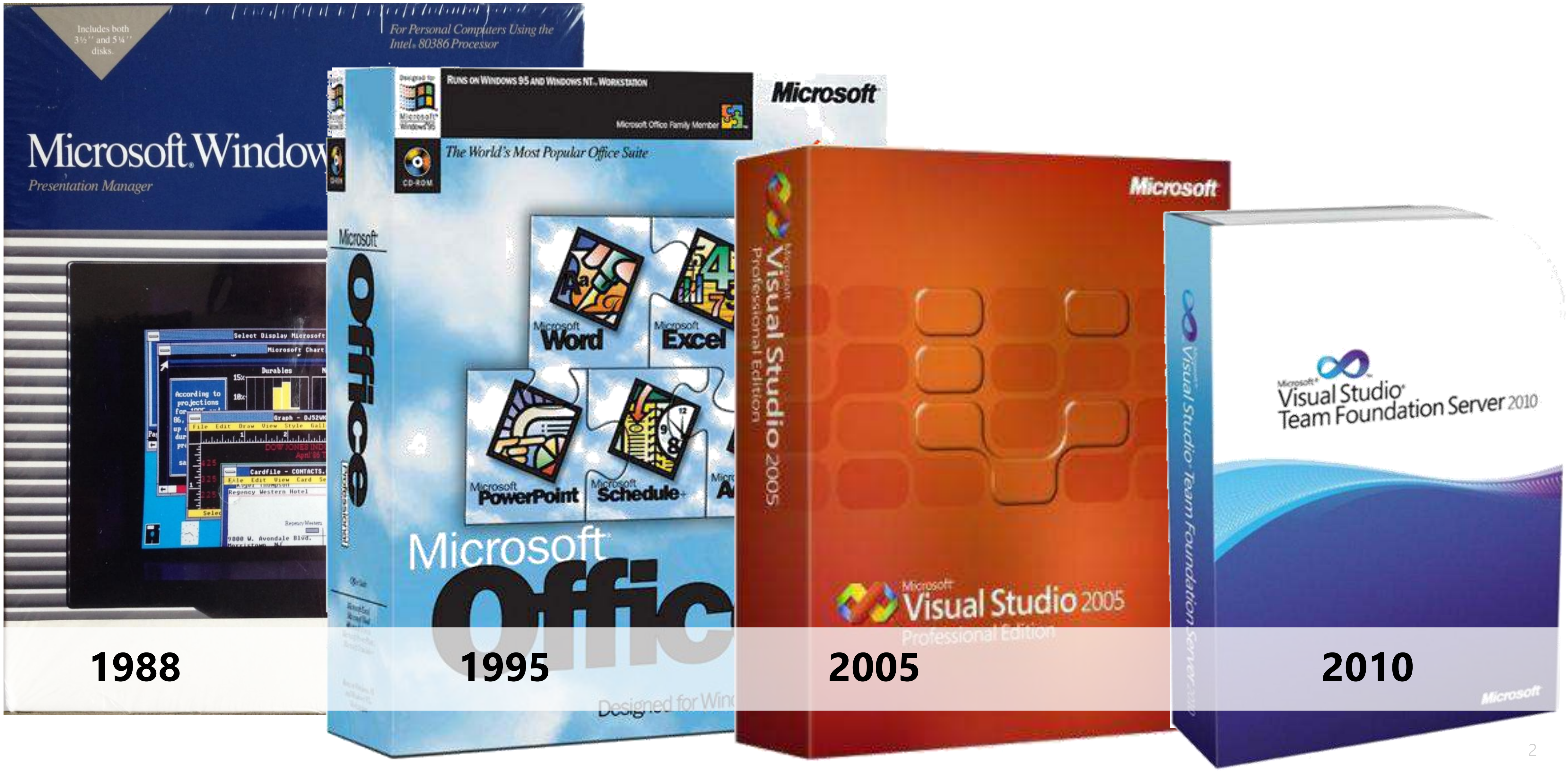
Microsoft

The DevOps Journey So Far at Microsoft

Kyle Krüsi
Senior Cloud Solution Architect
for Modern Service Management

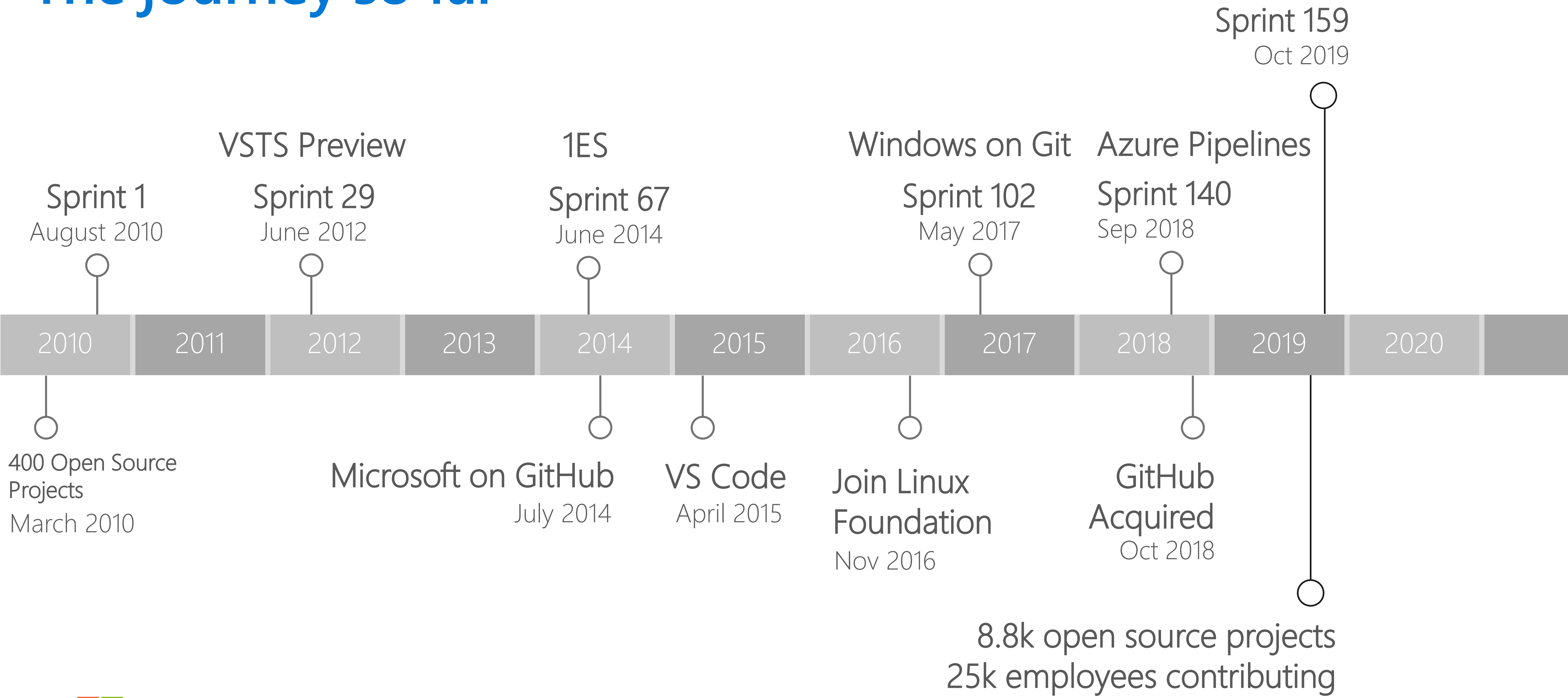


DevOps in Microsoft


Microsoft Windows
Presentation Manager
Microsoft Office
Microsoft Visual Studio 2005 Professional Edition
Microsoft Visual Studio Team Foundation Server 2010

1988 1995 2005 2010

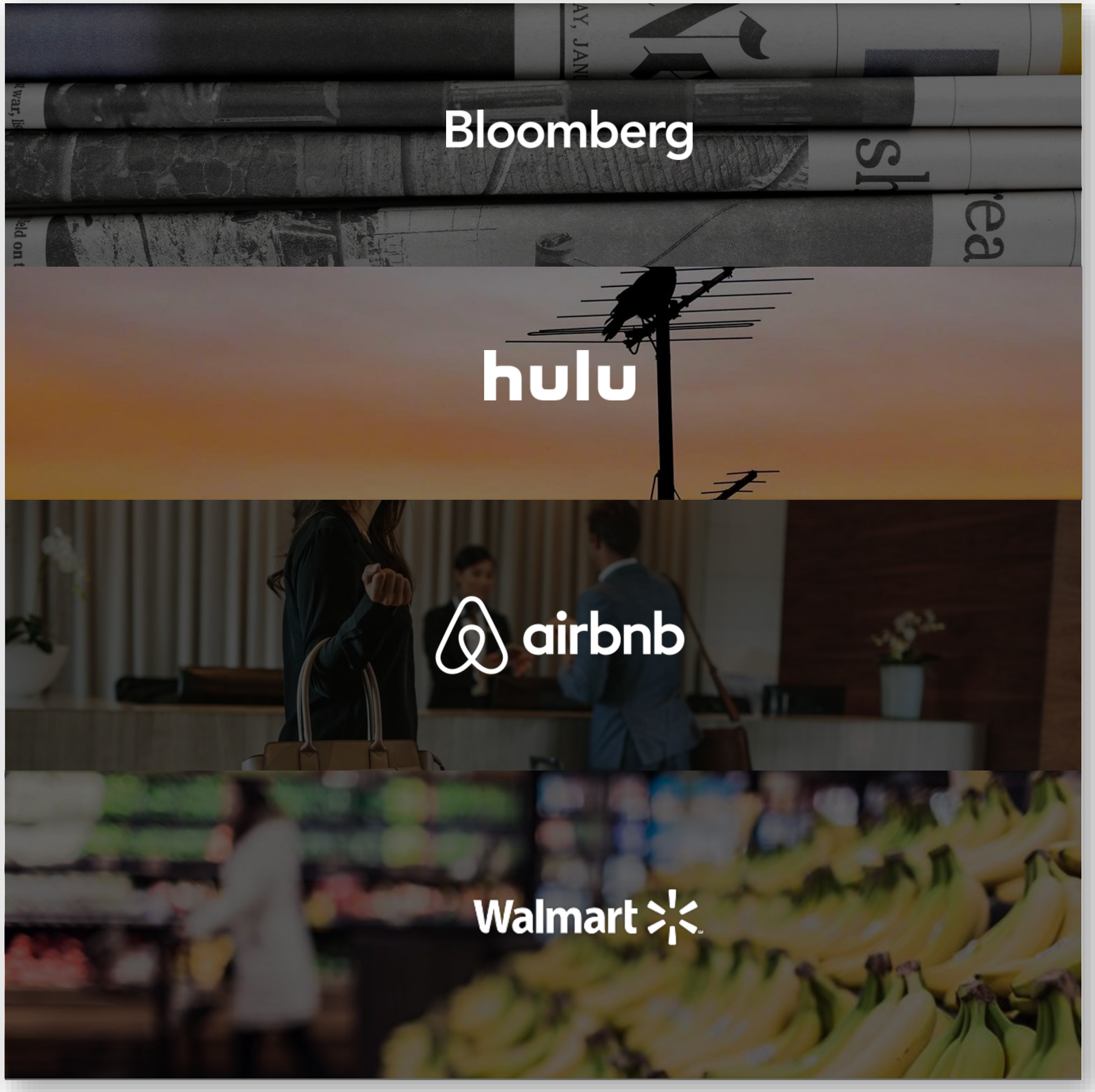
The journey so far



Microsoft

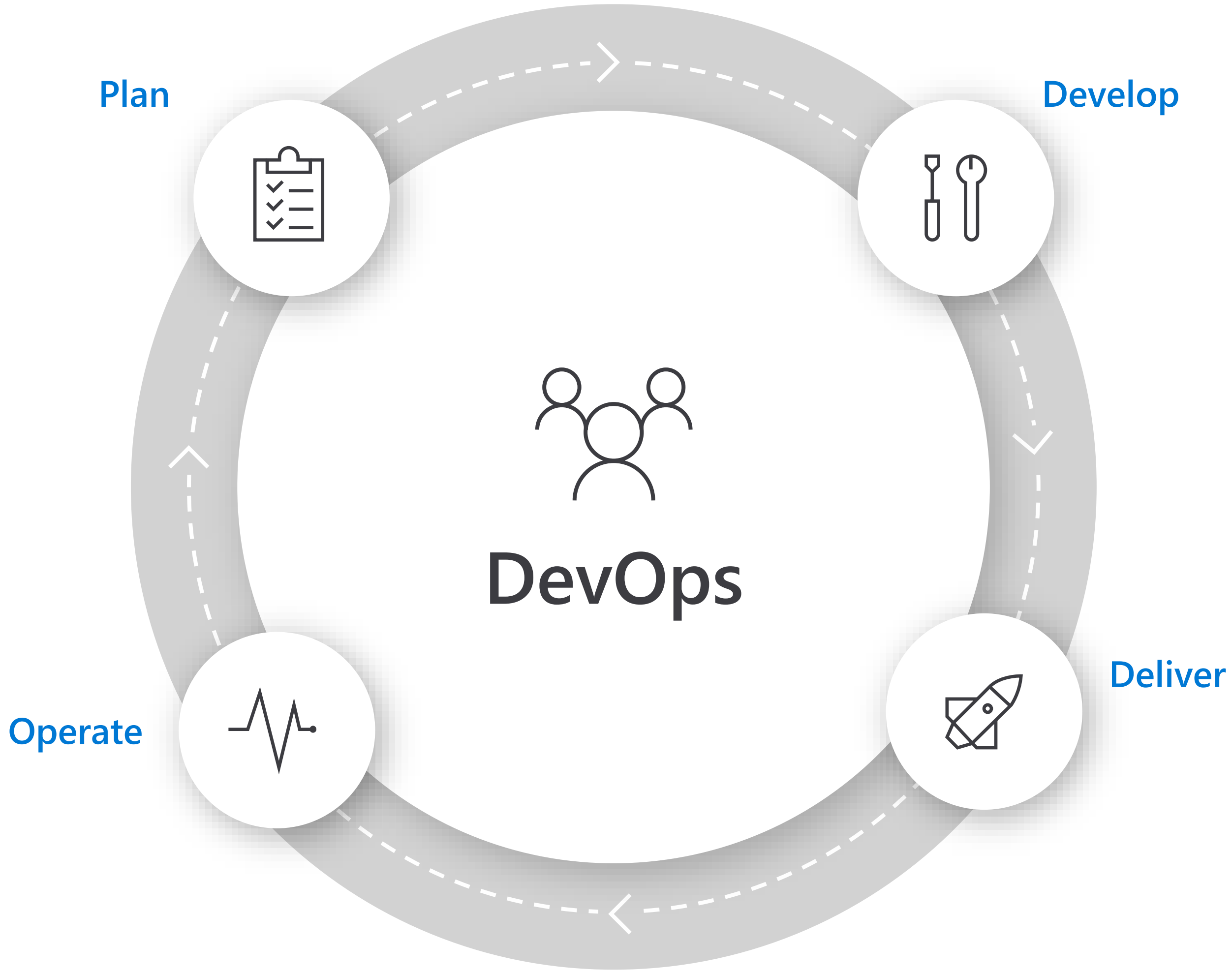
Every company is becoming a software company

62% of CEOs have an initiative to make their businesses more digital



Modern app engineering is enabled by DevOps

“ DevOps is the union of people, process, and technology to enable continuous delivery of value to your end users. ”



1ES using Azure DevOps and GitHub

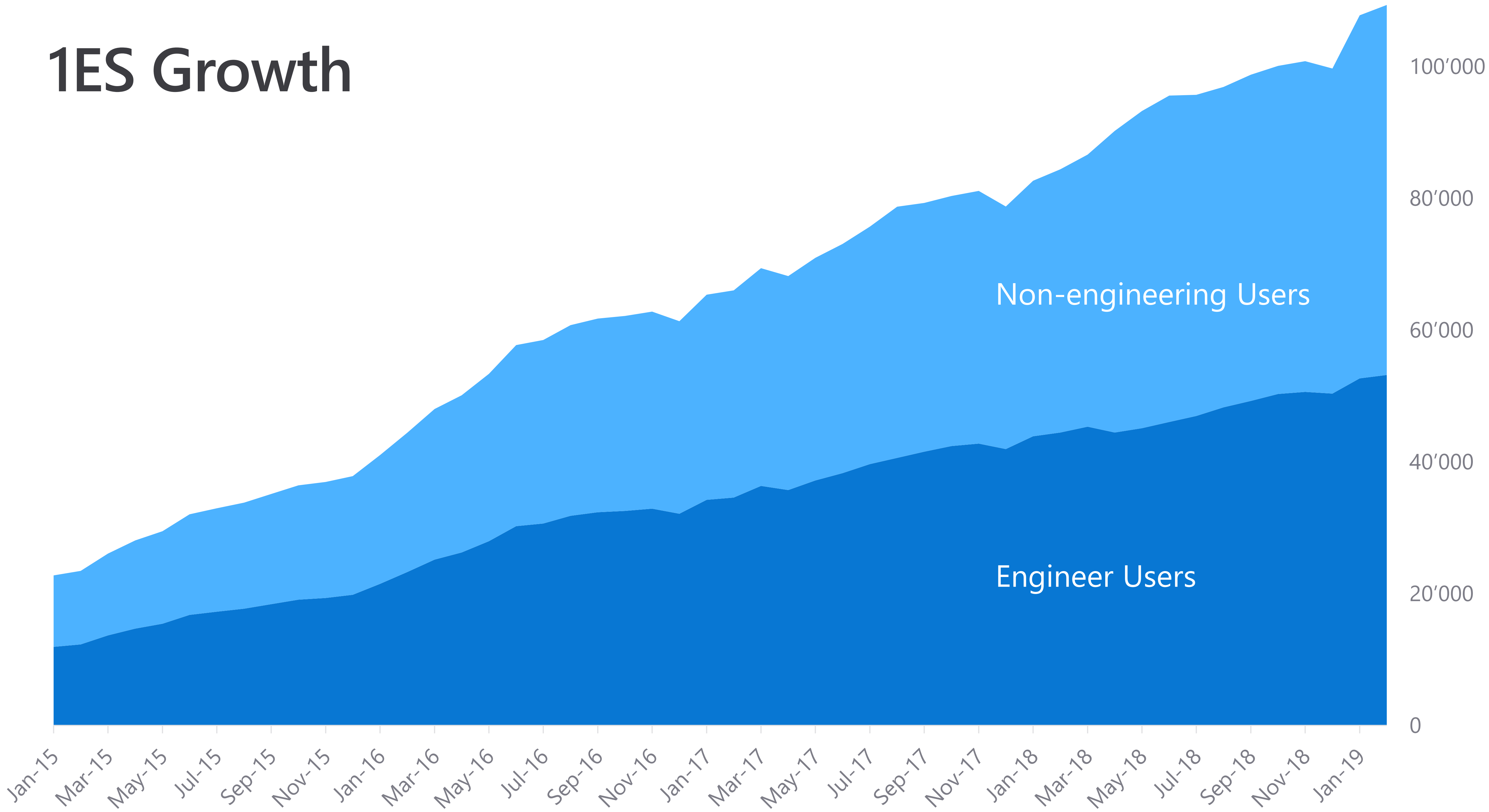
There cannot be a more important thing for an engineer, for a product team, than to work on the systems that drive our productivity.

So I would, any day of the week, trade off features for our own productivity.

I want our best engineers to work on our engineering systems, so that we can later on come back and build all of the new concepts we want.

- Satya Nadella

1ES Growth



Non-engineering Users
Engineer Users
100'000
80'000
60'000
40'000
20'000
0
Jan-15
Mar-15
May-15
Jul-15
Sep-15
Nov-15
Jan-16
Mar-16
May-16
Jul-16
Sep-16
Nov-16
Jan-17
Mar-17
May-17
Jul-17
Sep-17
Nov-17
Jan-18
Mar-18
May-18
Jul-18
Sep-18
Nov-18
Jan-19

DevOps across Microsoft in 2021 (Sep)

https://aka.ms/DevOps-Stories

425K
Pull Requests completed per month

13.3M
Builds per month

1.9M
Work items created per month

4.7M
Git commits per month

27+k
Repos: GitHub & Azure Repos

162K
Monthly Engaged Users (MEU)

5,600,000
Releases in August, 2021

Internal Microsoft Engineering System Activity, September 2021

Habits we've learned so far at Microsoft

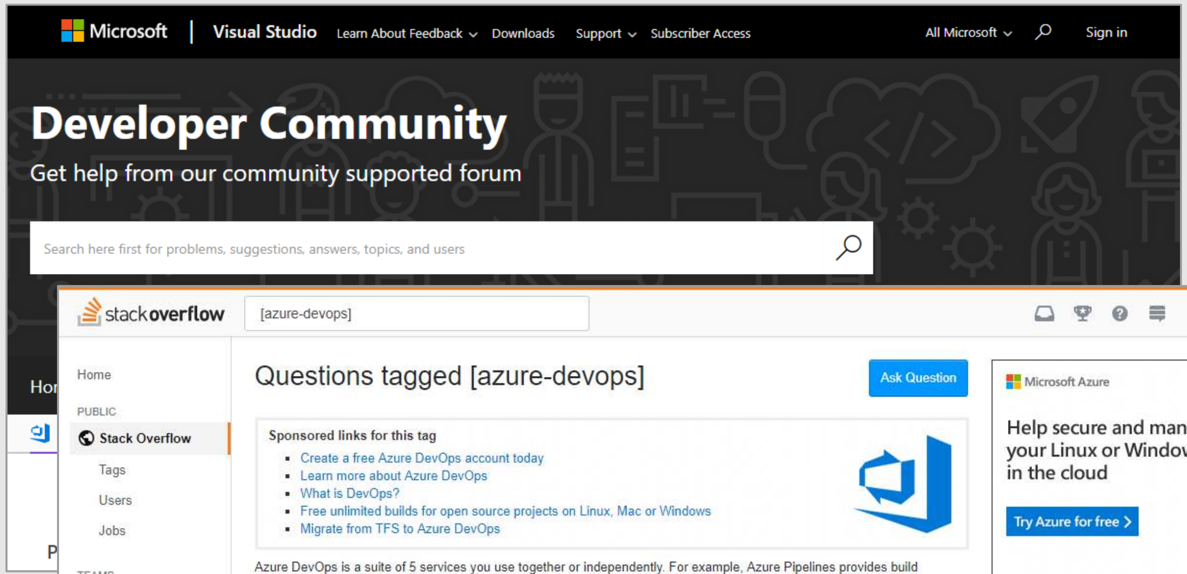
- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

Listen to our customers

Quantitively & Qualitatively

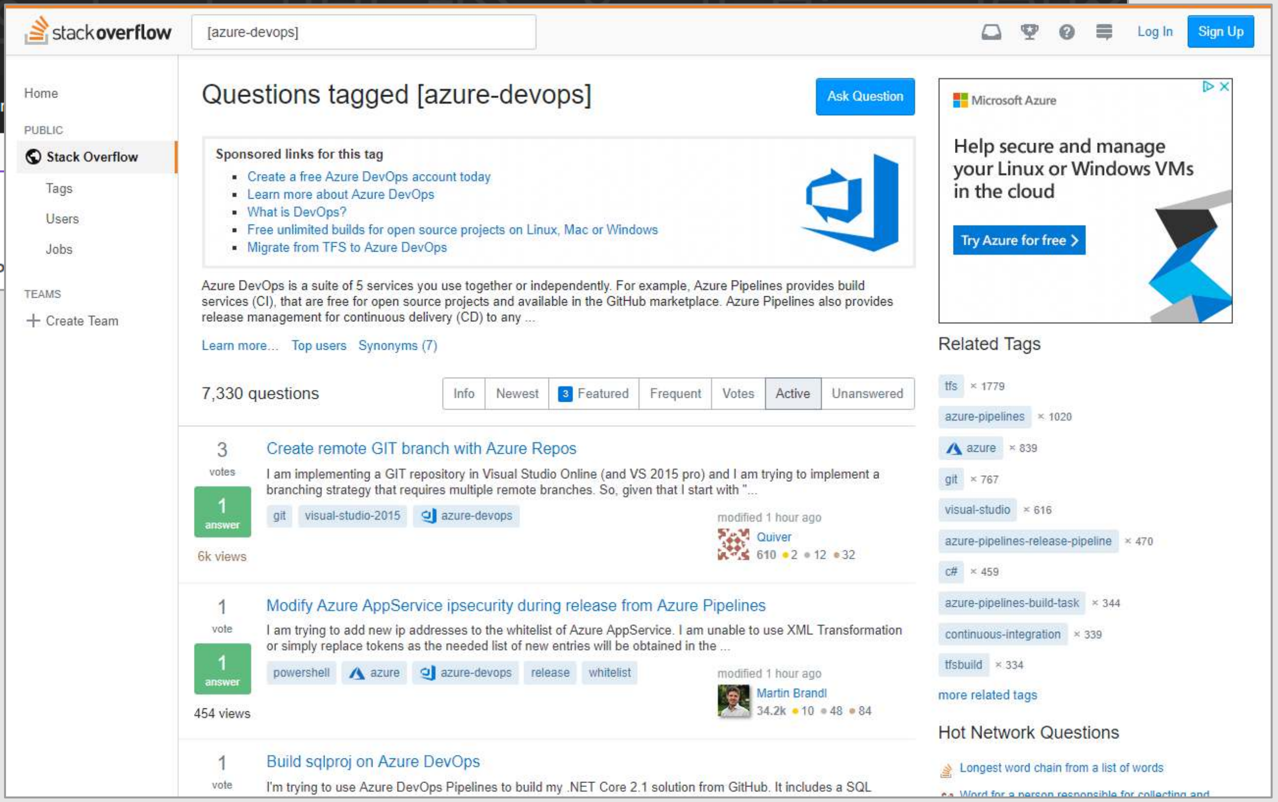


Microsoft | Visual Studio Learn About Feedback Downloads Support Subscriber Access All Microsoft Sign in

Developer Community

Get help from our community supported forum

Search here first for problems, suggestions, answers, topics, and users



stackoverflow [azure-devops] Log In Sign Up

Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS
Create Team

Questions tagged [azure-devops]

Ask Question

Sponsored links for this tag
- Create a free Azure DevOps account today
- Learn more about Azure DevOps
- What is DevOps?
- Free unlimited builds for open source projects on Linux, Mac or Windows
- Migrate from TFS to Azure DevOps

Azure DevOps is a suite of 5 services you use together or independently. For example, Azure Pipelines provides build services (CI), that are free for open source projects and available in the GitHub marketplace. Azure Pipelines also provides release management for continuous delivery (CD) to any ...

Learn more... Top users Synonyms (7)

7,330 questions Info Newest 3 Featured Frequent Votes Active Unanswered

3 votes, 1 answer, 6k views — **Create remote GIT branch with Azure Repos**
I am implementing a GIT repository in Visual Studio Online (and VS 2015 pro) and I am trying to implement a branching strategy that requires multiple remote branches. So, given that I start with "...
git visual-studio-2015 azure-devops — modified 1 hour ago Quiver 610 ●2 ●12 ●32

1 vote, 1 answer, 454 views — **Modify Azure AppService ipsecurity during release from Azure Pipelines**
I am trying to add new ip addresses to the whitelist of Azure AppService. I am unable to use XML Transformation or simply replace tokens as the needed list of new entries will be obtained in the ...
powershell azure azure-devops release whitelist — modified 1 hour ago Martin Brandl 34.2k ●10 ●48 ●84

1 vote — **Build sqlproj on Azure DevOps**
I'm trying to use Azure DevOps Pipelines to build my .NET Core 2.1 solution from GitHub. It includes a SQL

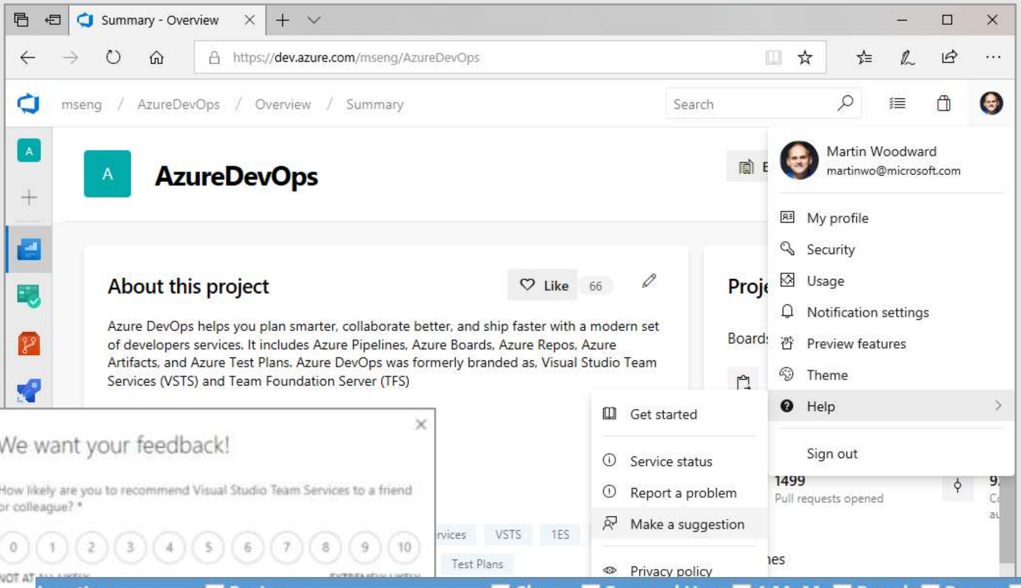
Microsoft Azure — Help secure and manage your Linux or Windows VMs in the cloud — Try Azure for free >

Related Tags

tfs	× 1779
azure-pipelines	× 1020
azure	× 839
git	× 767
visual-studio	× 616
azure-pipelines-release-pipeline	× 470
c#	× 459
azure-pipelines-build-task	× 344
continuous-integration	× 339
tfsbuild	× 334

more related tags

Hot Network Questions
- Longest word chain from a list of words
- Word for a person responsible for collecting and



Summary - Overview https://dev.azure.com/mseng/AzureDevOps

mseng / AzureDevOps / Overview / Summary Search

AzureDevOps

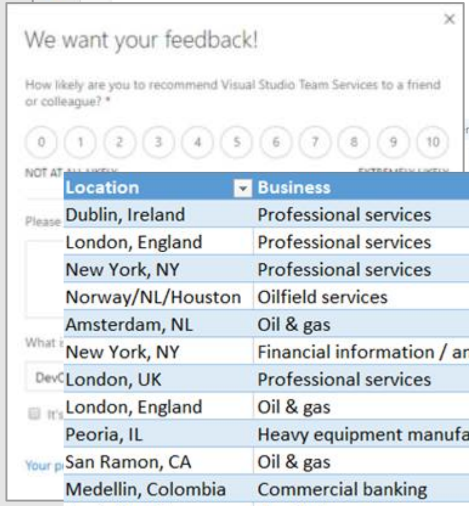
About this project Like 66

Azure DevOps helps you plan smarter, collaborate better, and ship faster with a modern set of developers services. It includes Azure Pipelines, Azure Boards, Azure Repos, Azure Artifacts, and Azure Test Plans. Azure DevOps was formerly branded as, Visual Studio Team Services (VSTS) and Team Foundation Server (TFS)

Martin Woodward martinwo@microsoft.com
My profile
Security
Usage
Notification settings
Preview features
Theme
Help
Sign out

Get started
Service status
Report a problem
Make a suggestion
Privacy policy

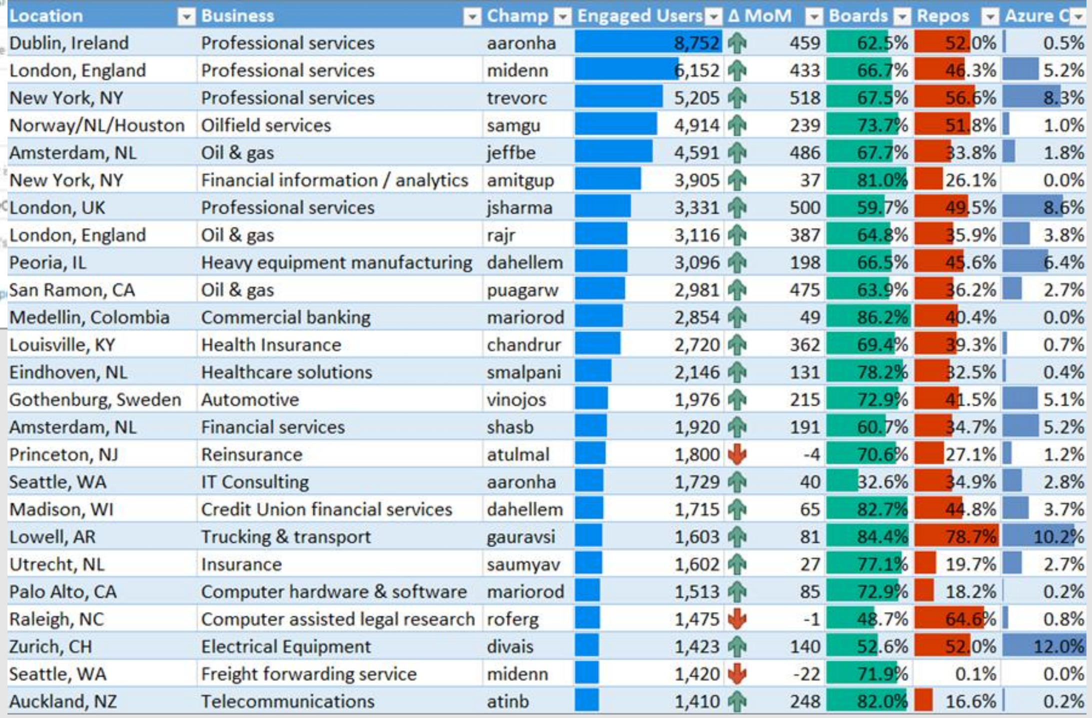
1499 Pull requests opened



We want your feedback!

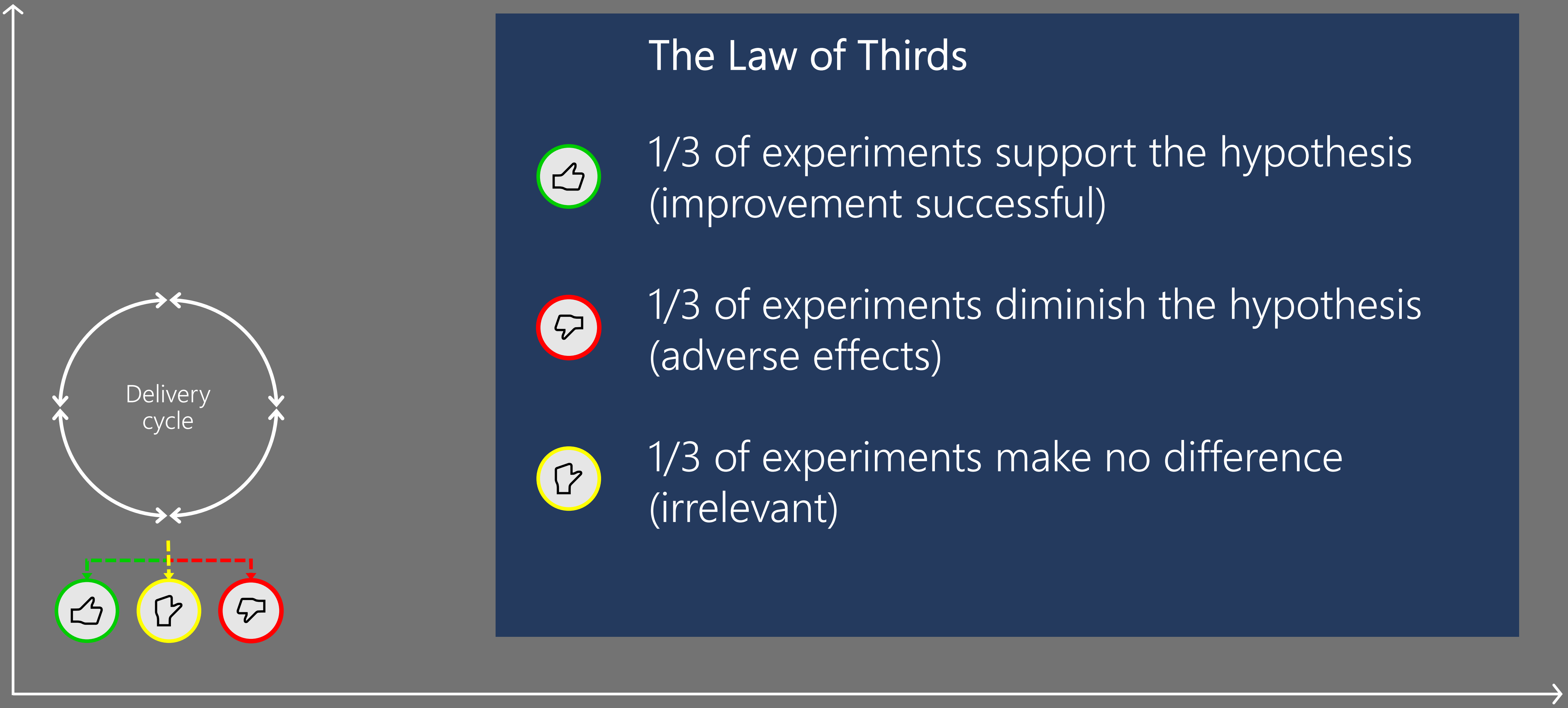
How likely are you to recommend Visual Studio Team Services to a friend or colleague? *

0 1 2 3 4 5 6 7 8 9 10



Location	Business	Champ	Engaged Users	Δ MoM	Boards	Repos	Azure C
Dublin, Ireland	Professional services	aaronha	8,752	459	62.5%	52.0%	0.5%
London, England	Professional services	midenn	6,152	433	66.7%	46.3%	5.2%
New York, NY	Professional services	trevorc	5,205	518	67.5%	56.6%	8.3%
Norway/NL/Houston	Oilfield services	samgu	4,914	239	73.7%	51.8%	1.0%
Amsterdam, NL	Oil & gas	jeffbe	4,591	486	67.7%	33.8%	1.8%
New York, NY	Financial information / analytics	amitgup	3,905	37	81.0%	26.1%	0.0%
London, UK	Professional services	jsharma	3,331	500	59.7%	49.5%	8.6%
London, England	Oil & gas	rajr	3,116	387	64.8%	35.9%	3.8%
Peoria, IL	Heavy equipment manufacturing	dahellem	3,096	198	66.5%	45.6%	6.4%
San Ramon, CA	Oil & gas	puagarw	2,981	475	63.9%	36.2%	2.7%
Medellin, Colombia	Commercial banking	mariorod	2,854	49	86.2%	40.4%	0.0%
Louisville, KY	Health Insurance	chandrur	2,720	362	69.4%	39.3%	0.7%
Eindhoven, NL	Healthcare solutions	smalpani	2,146	131	78.2%	32.5%	0.4%
Gothenburg, Sweden	Automotive	vinojos	1,976	215	72.9%	41.5%	5.1%
Amsterdam, NL	Financial services	shasb	1,920	191	60.7%	34.7%	5.2%
Princeton, NJ	Reinsurance	atulmal	1,800	-4	70.6%	27.1%	1.2%
Seattle, WA	IT Consulting	aaronha	1,729	40	32.6%	34.9%	2.8%
Madison, WI	Credit Union financial services	dahellem	1,715	65	82.7%	44.8%	3.7%
Lowell, AR	Trucking & transport	gauravsi	1,603	81	84.4%	78.7%	10.2%
Utrecht, NL	Insurance	saumyav	1,602	27	77.1%	19.7%	2.7%
Palo Alto, CA	Computer hardware & software	mariorod	1,513	85	72.9%	18.2%	0.2%
Raleigh, NC	Computer assisted legal research	roferg	1,475	-1	48.7%	64.6%	0.8%
Zurich, CH	Electrical Equipment	divais	1,423	140	52.6%	52.0%	12.0%
Seattle, WA	Freight forwarding service	midenn	1,420	-22	71.9%	0.1%	0.0%
Auckland, NZ	Telecommunications	atinb	1,410	248	82.0%	16.6%	0.2%

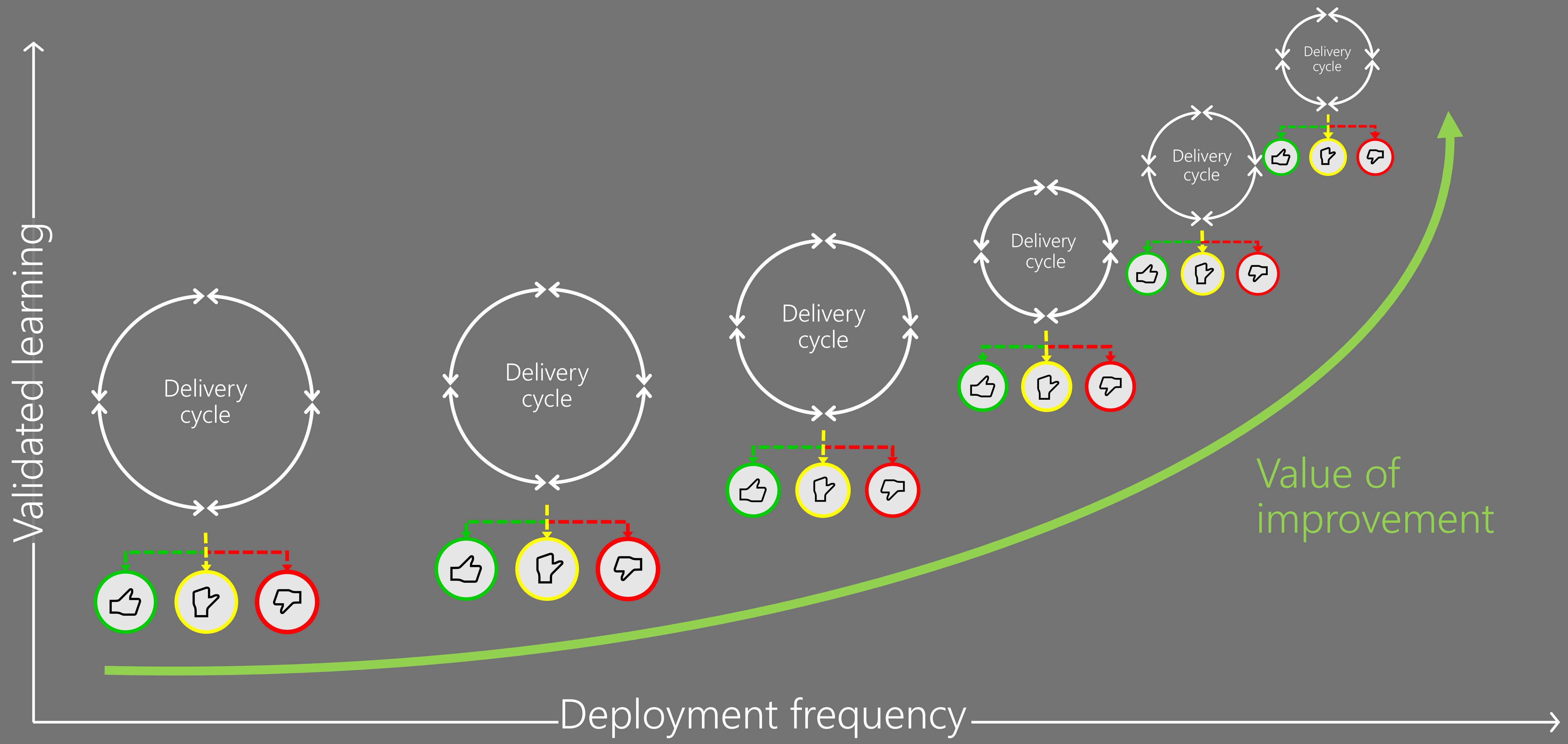
Ship to Learn



The Law of Thirds

- 👍 1/3 of experiments support the hypothesis (improvement successful)
- 👎 1/3 of experiments diminish the hypothesis (adverse effects)
- ✊ 1/3 of experiments make no difference (irrelevant)

Learning Accrues Compound Interest



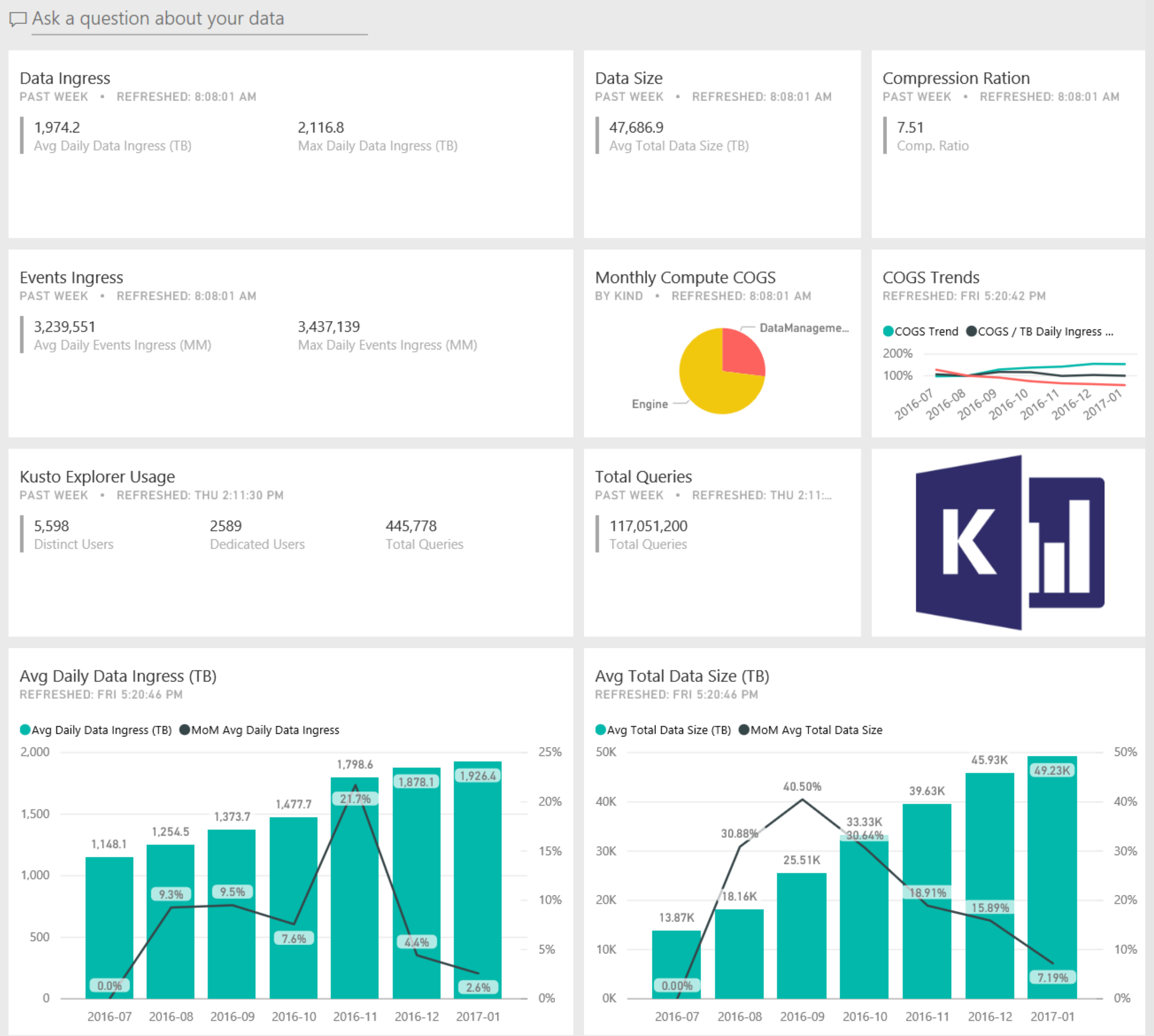
Our Definition of Done

Live in production, collecting telemetry supporting or diminishing the starting hypothesis.

Collect data broadly
(but carefully)

Application Insights Analytics (Project Kusto) for

- text search and queries over structured and semi-structured data
- high volume ingestion
- fast queries over very large data sets



Ask a question about your data

Data Ingress
PAST WEEK • REFRESHED: 8:08:01 AM
1,974.2 Avg Daily Data Ingress (TB)
2,116.8 Max Daily Data Ingress (TB)

Data Size
PAST WEEK • REFRESHED: 8:08:01 AM
47,686.9 Avg Total Data Size (TB)

Compression Ration
PAST WEEK • REFRESHED: 8:08:01 AM
7.51 Comp. Ratio

Events Ingress
PAST WEEK • REFRESHED: 8:08:01 AM
3,239,551 Avg Daily Events Ingress (MM)
3,437,139 Max Daily Events Ingress (MM)

Monthly Compute COGS
BY KIND • REFRESHED: 8:08:01 AM

COGS Trends
REFRESHED: FRI 5:20:42 PM

Kusto Explorer Usage
PAST WEEK • REFRESHED: THU 2:11:30 PM
5,598 Distinct Users
2589 Dedicated Users
445,778 Total Queries

Total Queries
PAST WEEK • REFRESHED: THU 2:11:...
117,051,200 Total Queries

Avg Daily Data Ingress (TB)
REFRESHED: FRI 5:20:46 PM

Avg Total Data Size (TB)
REFRESHED: FRI 5:20:46 PM

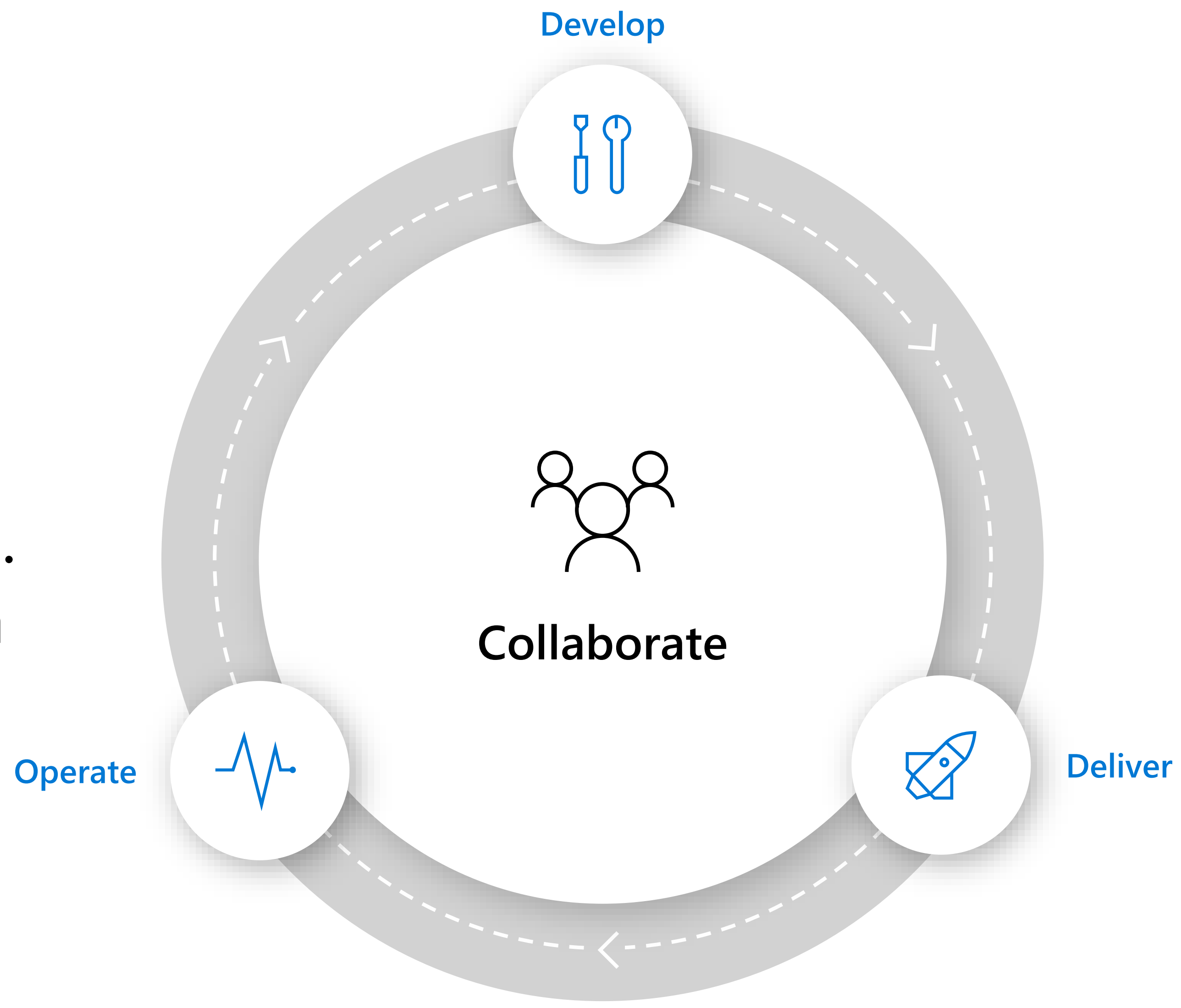
Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

Iterate over Pain

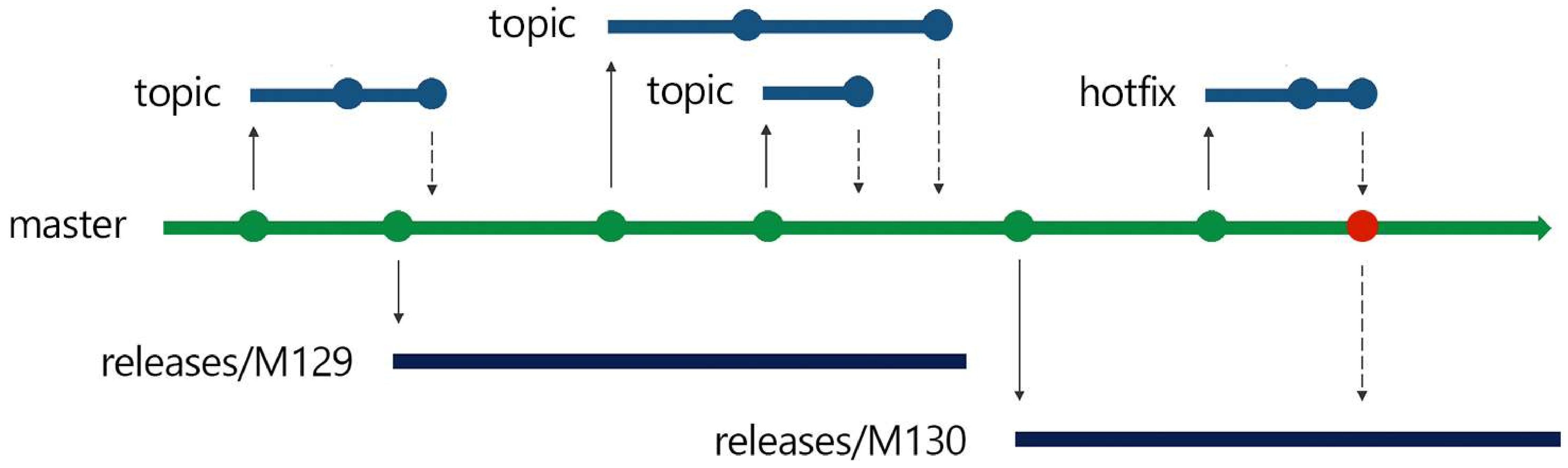
Find what hurts and keep doing it a bit better

"Find the part of your process in **getting value to customers** that slows you down or hurts the most. Make it **incrementally better** each sprint. Re-evaluate and improve the next most painful."



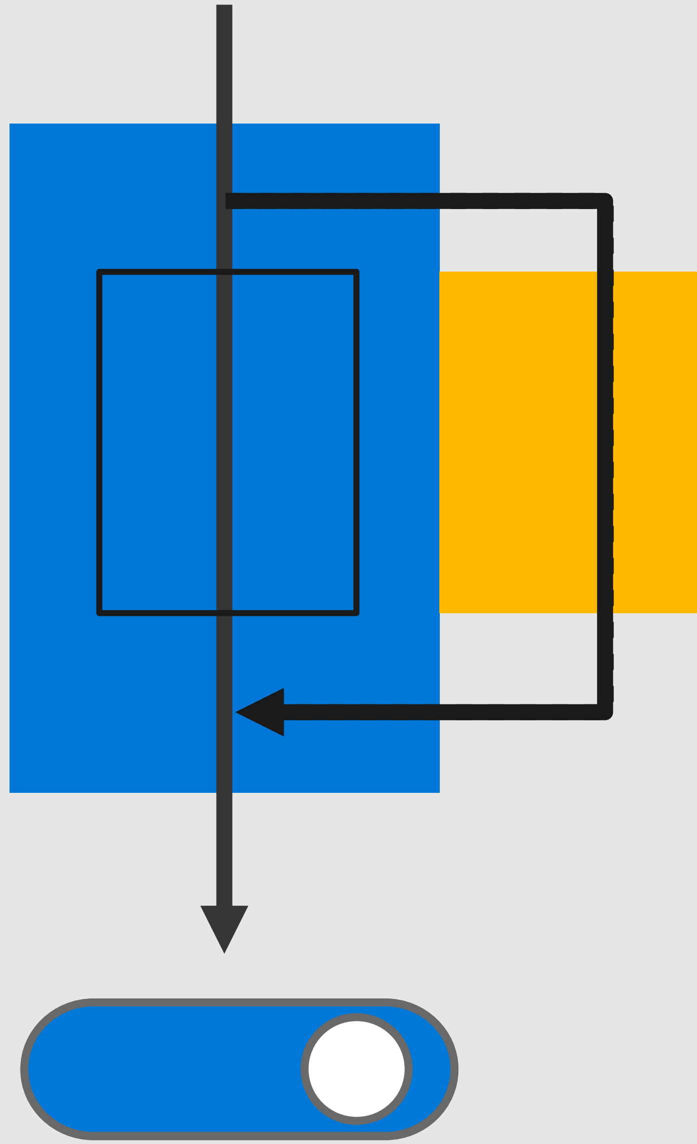
Release Flow

Using Trunk Based Development to avoid Merge Hell



Feature Flags

- All code is deployed, but feature flags control exposure
  - Reduces integration debt
- Flags provide runtime control down to individual user
- Users can be added or removed with no redeployment
- Mechanism for progressive experimentation & refinement
- Enables dark launch



Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- **Production First Mindset**
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

You Build It, You Love It

Every Live Site Incident (LSI) is a terrible thing to waste.

- All engineers rotate on call
- Detect>Mitigate>Remediate>Prevent
- Shameless postmortems weekly
- Closing LSI requires listing repair items
- LSI repair items in backlog with 2-sprint rule
- Alerts must be actionable
- Availability measured for each user per SLA
- Monthly service health review for kaizen

Live Site Incidents

On detection,

- Auto-alert on call DRI's
- Auto-create conference bridge
- Communicate externally and internally

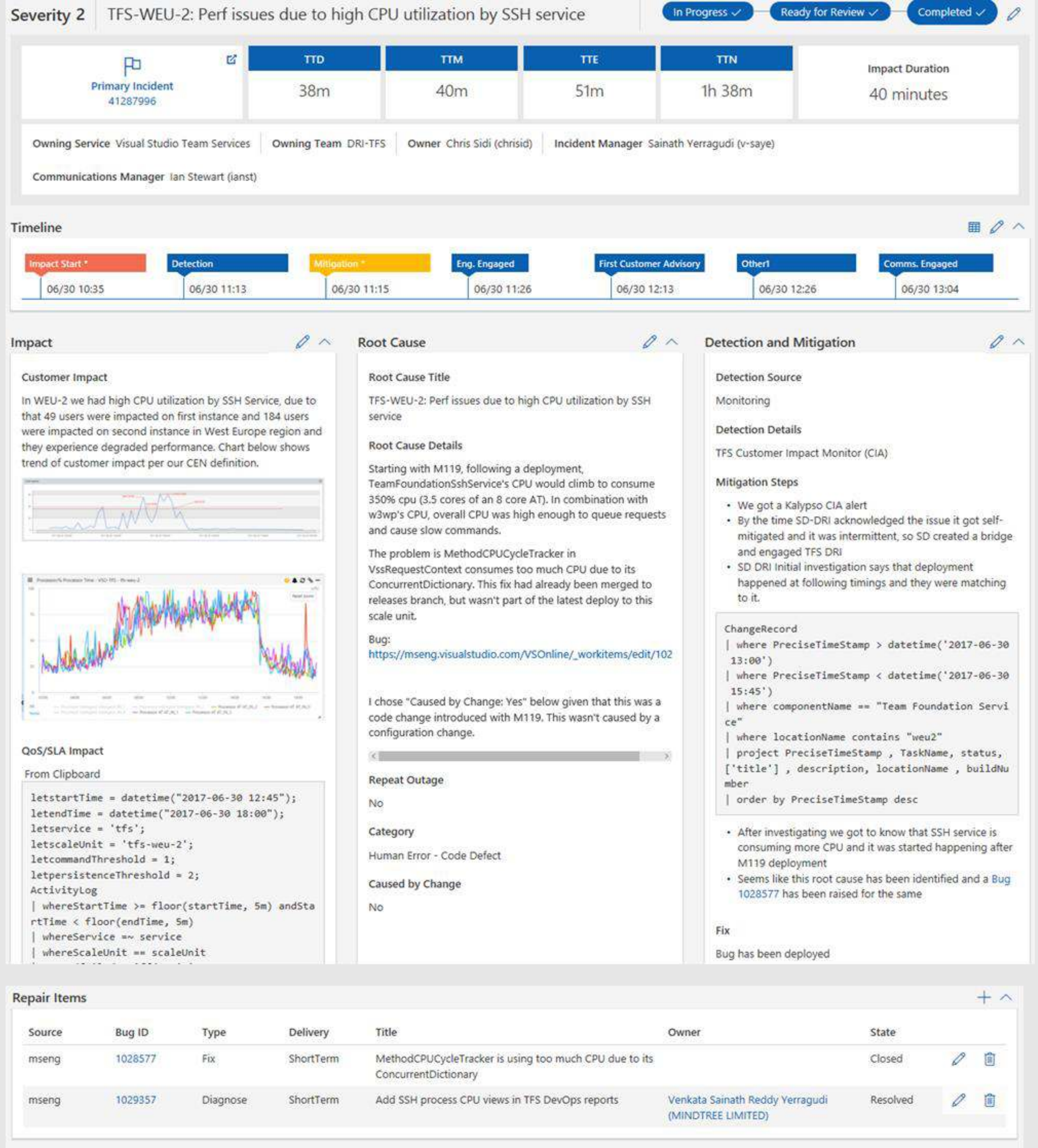
Gather data for repair items & mitigate for customers

Record every action

Use repair Items to prevent recurrence

- Create, track and timebox

Plan to rotate people during long LSI's



Severity 2 TFS-WEU-2: Perf issues due to high CPU utilization by SSH service

In Progress ✓ Ready for Review ✓ Completed ✓

Primary Incident 41287996

TTD	TTM	TTE	TTN	Impact Duration
38m	40m	51m	1h 38m	40 minutes

Owning Service Visual Studio Team Services | **Owning Team** DRI-TFS | **Owner** Chris Sidi (chrisid) | **Incident Manager** Sainath Yerragudi (v-saye)

Communications Manager Ian Stewart (ianst)

Timeline

Impact Start *	Detection	Mitigation *	Eng. Engaged	First Customer Advisory	Other1	Comms. Engaged
06/30 10:35	06/30 11:13	06/30 11:15	06/30 11:26	06/30 12:13	06/30 12:26	06/30 13:04

Impact

Customer Impact

In WEU-2 we had high CPU utilization by SSH Service, due to that 49 users were impacted on first instance and 184 users were impacted on second instance in West Europe region and they experience degraded performance. Chart below shows trend of customer impact per our CEN definition.

QoS/SLA Impact

From Clipboard

```
letstartTime = datetime("2017-06-30 12:45");
letendTime = datetime("2017-06-30 18:00");
letservice = 'tfs';
letscaleUnit = 'tfs-weu-2';
letcommandThreshold = 1;
letpersistenceThreshold = 2;
ActivityLog
| whereStartTime >= floor(startTime, 5m) andSta
rtTime < floor(endTime, 5m)
| whereService =~ service
| whereScaleUnit == scaleUnit
```

Root Cause

Root Cause Title

TFS-WEU-2: Perf issues due to high CPU utilization by SSH service

Root Cause Details

Starting with M119, following a deployment, TeamFoundationSshService's CPU would climb to consume 350% cpu (3.5 cores of an 8 core AT). In combination with w3wp's CPU, overall CPU was high enough to queue requests and cause slow commands.

The problem is MethodCPUCycleTracker in VssRequestContext consumes too much CPU due to its ConcurrentDictionary. This fix had already been merged to releases branch, but wasn't part of the latest deploy to this scale unit.

Bug:
https://mseng.visualstudio.com/VSOnline/_workitems/edit/102

I chose "Caused by Change: Yes" below given that this was a code change introduced with M119. This wasn't caused by a configuration change.

Repeat Outage

No

Category

Human Error - Code Defect

Caused by Change

No

Detection and Mitigation

Detection Source

Monitoring

Detection Details

TFS Customer Impact Monitor (CIA)

Mitigation Steps

- We got a Kalypso CIA alert
- By the time SD-DRI acknowledged the issue it got self-mitigated and it was intermittent, so SD created a bridge and engaged TFS DRI
- SD DRI Initial investigation says that deployment happened at following timings and they were matching to it.

```
ChangeRecord
| where PreciseTimeStamp > datetime('2017-06-30
 13:00')
| where PreciseTimeStamp < datetime('2017-06-30
 15:45')
| where componentName == "Team Foundation Servi
ce"
| where locationName contains "weu2"
| project PreciseTimeStamp , TaskName, status,
['title'] , description, locationName , buildNu
mber
| order by PreciseTimeStamp desc
```

- After investigating we got to know that SSH service is consuming more CPU and it was started happening after M119 deployment
- Seems like this root cause has been identified and a Bug 1028577 has been raised for the same

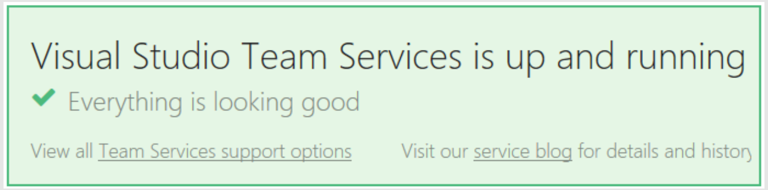
Fix

Bug has been deployed

Repair Items

Source	Bug ID	Type	Delivery	Title	Owner	State
mseng	1028577	Fix	ShortTerm	MethodCPUCycleTracker is using too much CPU due to its ConcurrentDictionary		Closed
mseng	1029357	Diagnose	ShortTerm	Add SSH process CPU views in TFS DevOps reports	Venkata Sainath Reddy Yerragudi (MINDTREE LIMITED)	Resolved

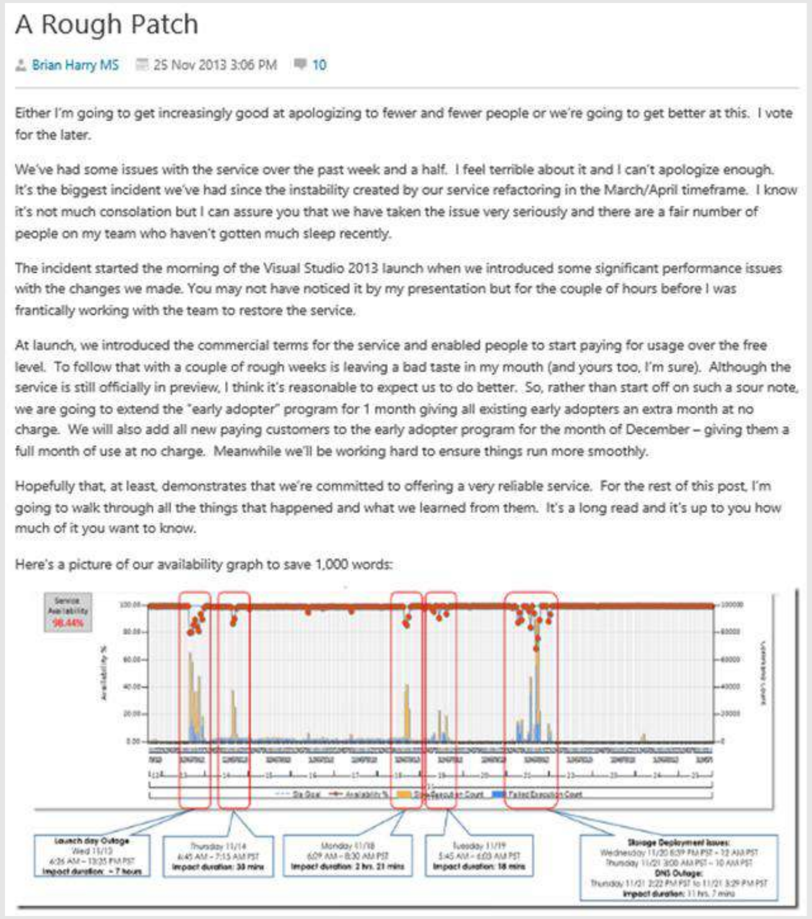
Be Transparent



Visual Studio Team Services is up and running

✓ Everything is looking good

View all Team Services support options   Visit our service blog for details and histor



A Rough Patch

Brian Harry MS 25 Nov 2013 3:06 PM 10

Either I'm going to get increasingly good at apologizing to fewer and fewer people or we're going to get better at this. I vote for the later.

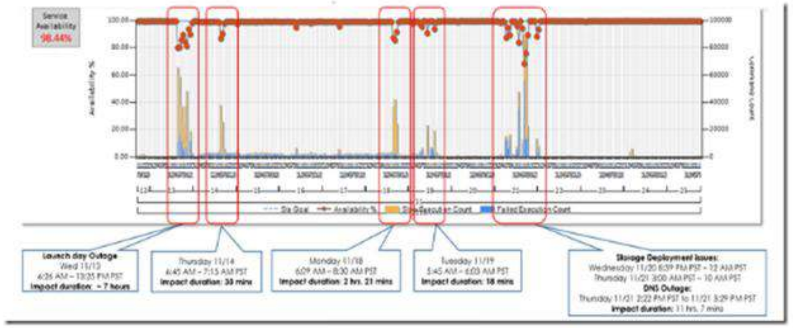
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you that we have taken the issue very seriously and there are a fair number of people on my team who haven't gotten much sleep recently.

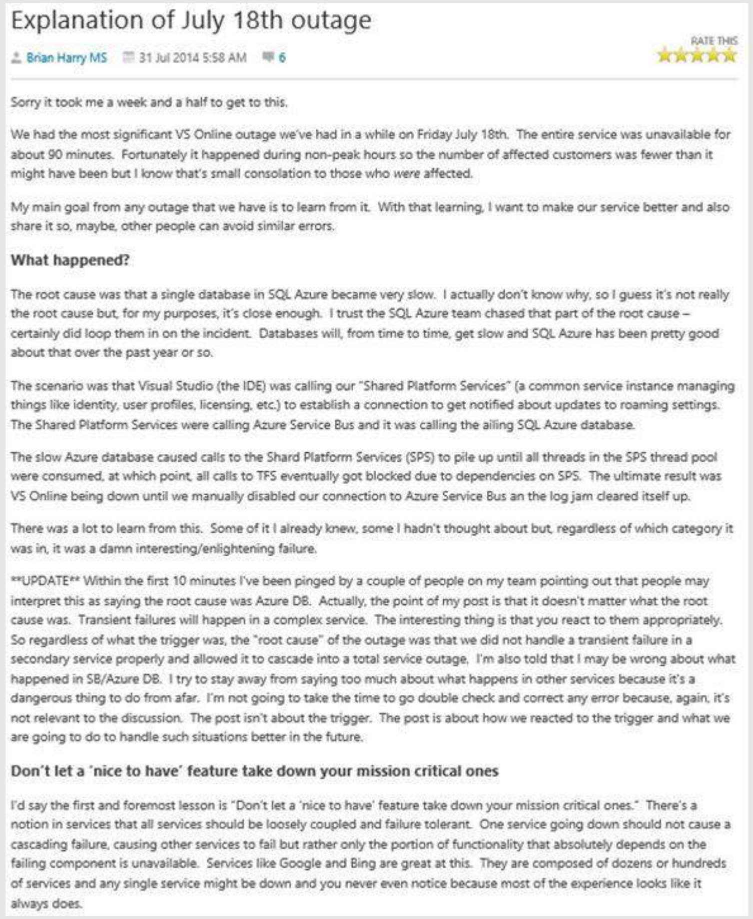
The incident started the morning of the Visual Studio 2013 launch when we introduced some significant performance issues with the changes we made. You may not have noticed it by my presentation but for the couple of hours before I was frantically working with the team to restore the service.

At launch, we introduced the commercial terms for the service and enabled people to start paying for usage over the free level. To follow that with a couple of rough weeks is leaving a bad taste in my mouth (and yours too, I'm sure). Although the service is still officially in preview, I think it's reasonable to expect us to do better. So, rather than start off on such a sour note, we are going to extend the "early adopter" program for 1 month giving all existing early adopters an extra month at no charge. We will also add all new paying customers to the early adopter program for the month of December – giving them a full month of use at no charge. Meanwhile we'll be working hard to ensure things run more smoothly.

Hopefully that, at least, demonstrates that we're committed to offering a very reliable service. For the rest of this post, I'm going to walk through all the things that happened and what we learned from them. It's a long read and it's up to you how much of it you want to know.

Here's a picture of our availability graph to save 1,000 words:





Explanation of July 18th outage

Brian Harry MS 31 Jul 2014 5:58 AM 6

RATE THIS ★★★★★

Sorry it took me a week and a half to get to this.

We had the most significant VS Online outage we've had in a while on Friday July 18th. The entire service was unavailable for about 90 minutes. Fortunately it happened during non-peak hours so the number of affected customers was fewer than it might have been but I know that's small consolation to those who *were* affected.

My main goal from any outage that we have is to learn from it. With that learning, I want to make our service better and also share it so, maybe, other people can avoid similar errors.

What happened?

The root cause was that a single database in SQL Azure became very slow. I actually don't know why, so I guess it's not really the root cause but, for my purposes, it's close enough. I trust the SQL Azure team chased that part of the root cause – certainly did loop them in on the incident. Databases will, from time to time, get slow and SQL Azure has been pretty good about that over the past year or so.

The scenario was that Visual Studio (the IDE) was calling our "Shared Platform Services" (a common service instance managing things like identity, user profiles, licensing, etc.) to establish a connection to get notified about updates to roaming settings. The Shared Platform Services were calling Azure Service Bus and it was calling the ailing SQL Azure database.

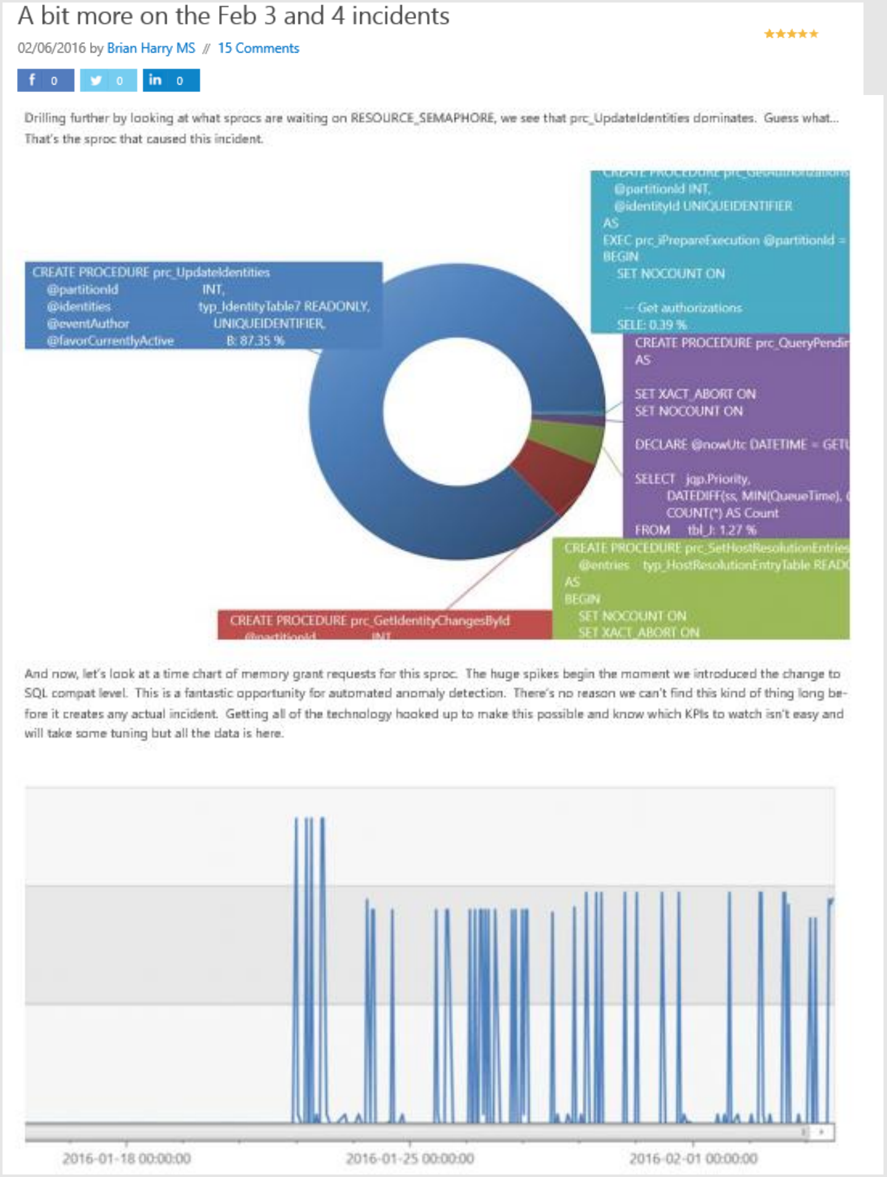
The slow Azure database caused calls to the Shard Platform Services (SPS) to pile up until all threads in the SPS thread pool were consumed, at which point, all calls to TFS eventually got blocked due to dependencies on SPS. The ultimate result was VS Online being down until we manually disabled our connection to Azure Service Bus an the log jam cleared itself up.

There was a lot to learn from this. Some of it I already knew, some I hadn't thought about but, regardless of which category it was in, it was a damn interesting/enlightening failure.

UPDATE Within the first 10 minutes I've been pinged by a couple of people on my team pointing out that people may interpret this as saying the root cause was Azure DB. Actually, the point of my post is that it doesn't matter what the root cause was. Transient failures will happen in a complex service. The interesting thing is that you react to them appropriately. So regardless of what the trigger was, the "root cause" of the outage was that we did not handle a transient failure in a secondary service properly and allowed it to cascade into a total service outage. I'm also told that I may be wrong about what happened in SB/Azure DB. I try to stay away from saying too much about what happens in other services because it's a dangerous thing to do from afar. I'm not going to take the time to go double check and correct any error because, again, it's not relevant to the discussion. The post isn't about the trigger. The post is about how we reacted to the trigger and what we are going to do to handle such situations better in the future.

Don't let a 'nice to have' feature take down your mission critical ones

I'd say the first and foremost lesson is "Don't let a 'nice to have' feature take down your mission critical ones." There's a notion in services that all services should be loosely coupled and failure tolerant. One service going down should not cause a cascading failure, causing other services to fail but rather only the portion of functionality that absolutely depends on the failing component is unavailable. Services like Google and Bing are great at this. They are composed of dozens or hundreds of services and any single service might be down and you never even notice because most of the experience looks like it always does.



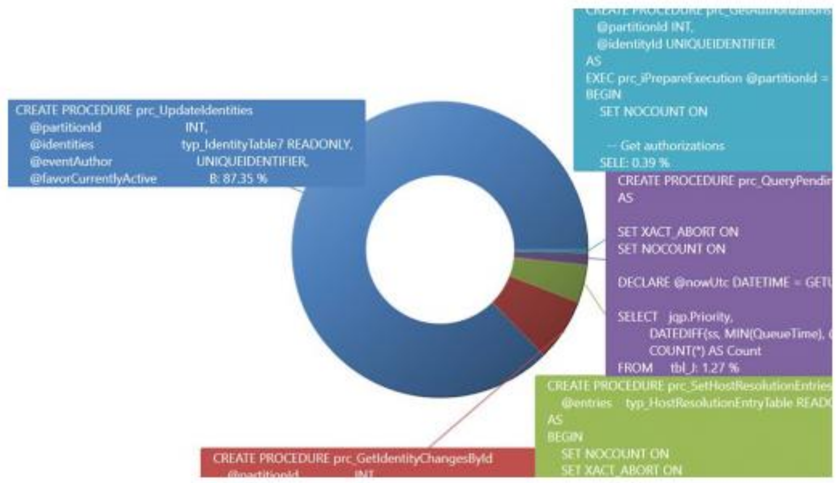
A bit more on the Feb 3 and 4 incidents

02/06/2016 by Brian Harry MS // 15 Comments

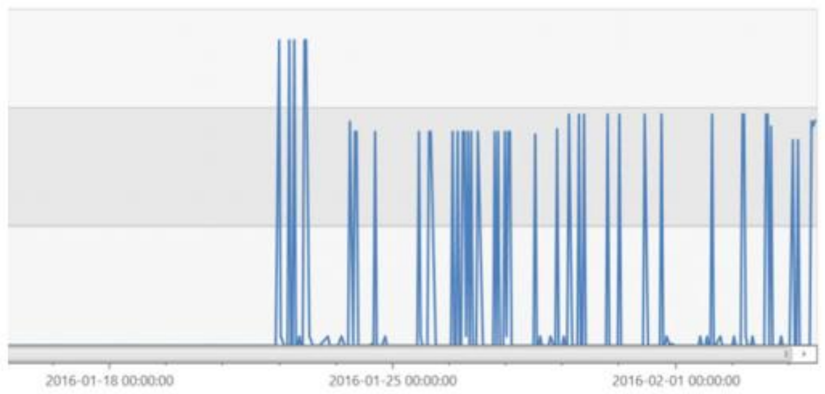
★★★★★

f 0 0 in 0

Drilling further by looking at what sprocs are waiting on RESOURCE_SEMAPHORE, we see that prc_UpdateIdentities dominates. Guess what… That's the sproc that caused this incident.

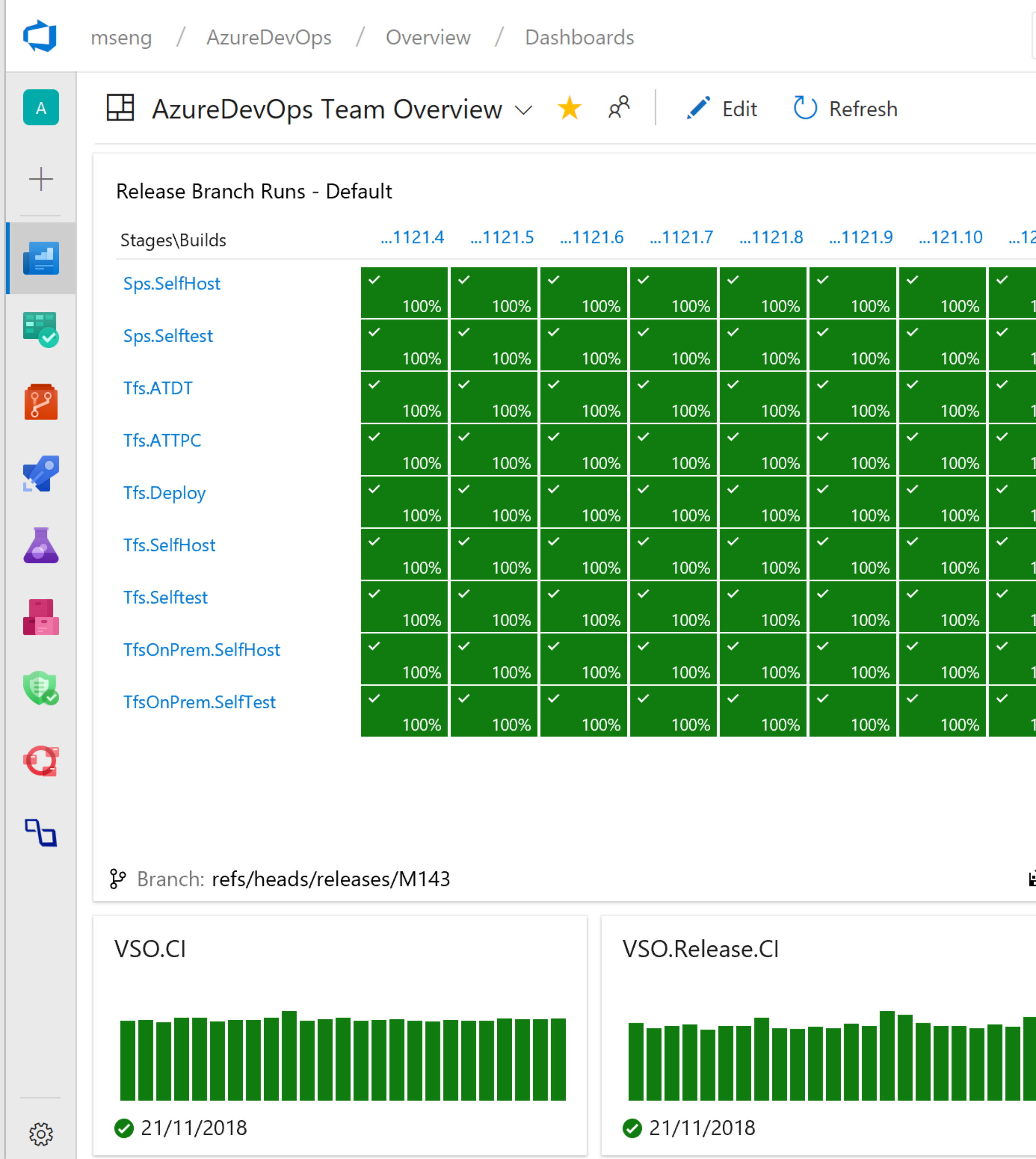


And now, let's look at a time chart of memory grant requests for this sproc. The huge spikes begin the moment we introduced the change to SQL compat level. This is a fantastic opportunity for automated anomaly detection. There's no reason we can't find this kind of thing long before it creates any actual incident. Getting all of the technology hooked up to make this possible and know which KPIs to watch isn't easy and will take some tuning but all the data is here.



Automate completely

- No such thing as 'partial automation'
- No more "one time" commands run manually
- Every command goes in PowerShell scripts that are checked in
- Deployment to pre-production & canary is the same as deployment to production **every time**
- All orchestrated with Azure Pipelines


mseng / AzureDevOps / Overview / Dashboards
A
AzureDevOps Team Overview
Edit
Refresh

Release Branch Runs - Default

Stages\Builds	...1121.4	...1121.5	...1121.6	...1121.7	...1121.8	...1121.9	...121.10	...12
Sps.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Sps.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.ATDT	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.ATTPC	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
Tfs.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 1

Branch: refs/heads/releases/M143

VSO.CI

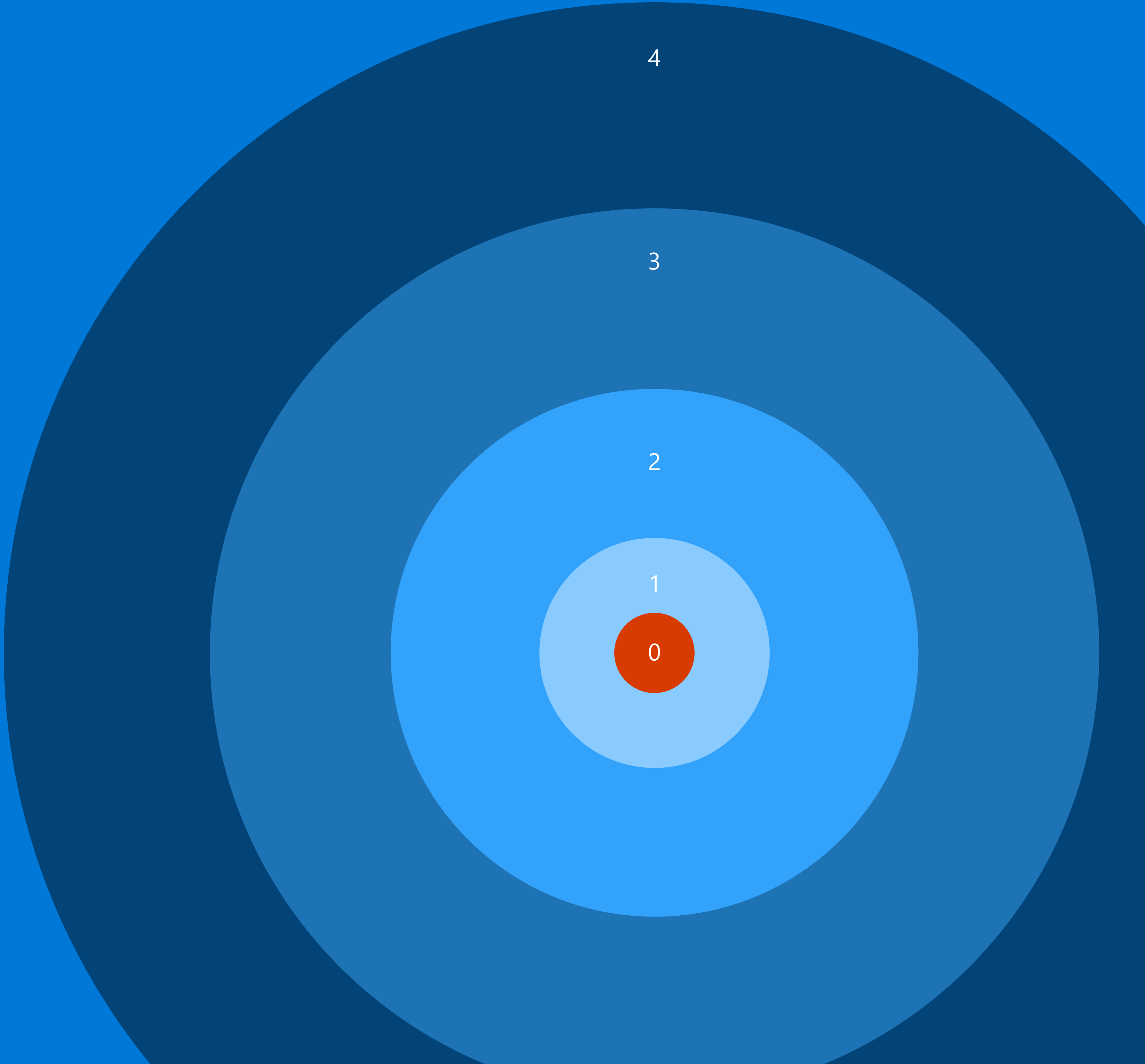
✓ 21/11/2018

VSO.Release.CI

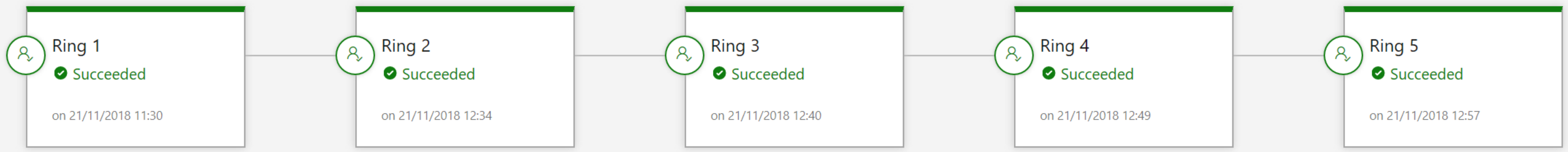
✓ 21/11/2018

Your aim won't be perfect.

Control the blast radius.



Tracking Deployments to Production (5 Rings)



Ring 1
✓ Succeeded
on 21/11/2018 11:30

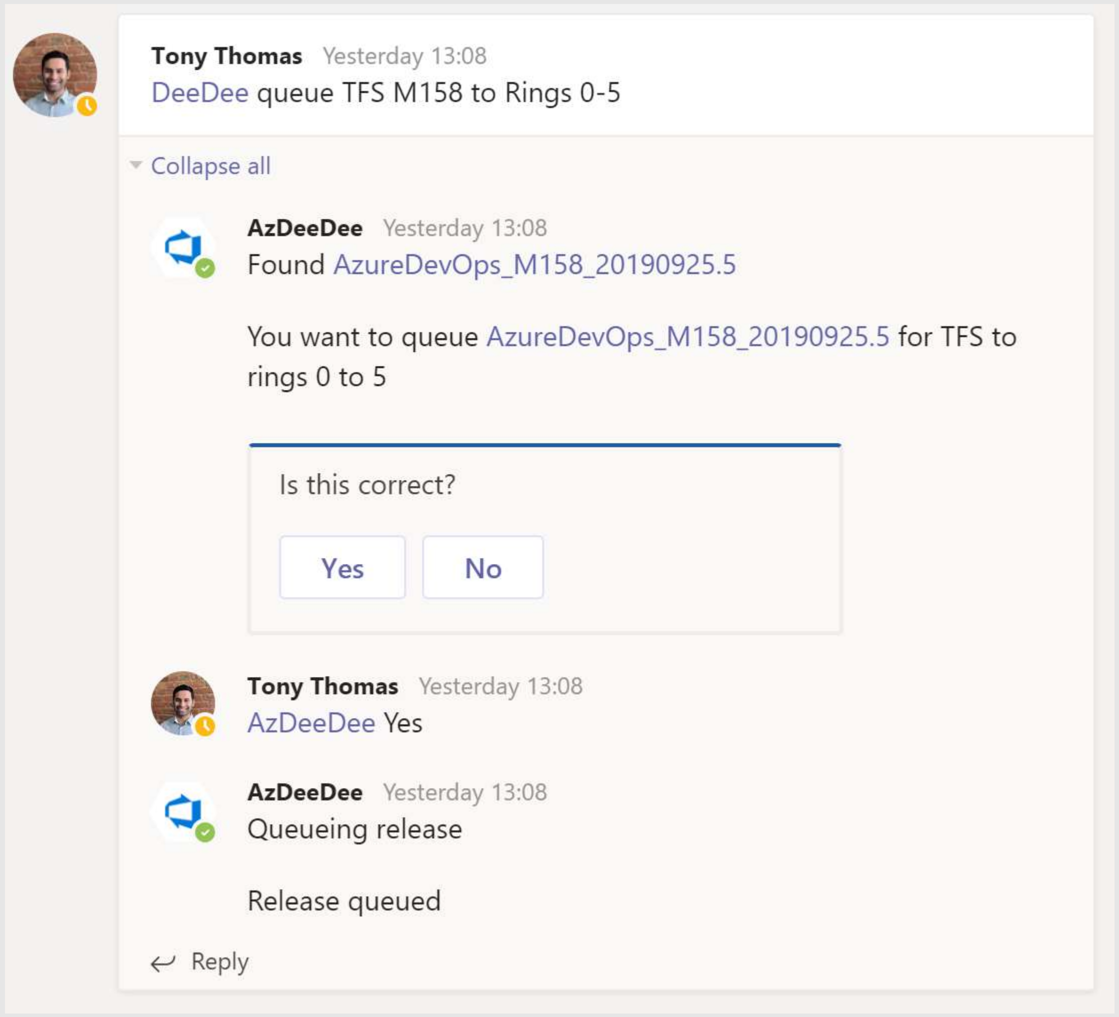
Ring 2
✓ Succeeded
on 21/11/2018 12:34

Ring 3
✓ Succeeded
on 21/11/2018 12:40

Ring 4
✓ Succeeded
on 21/11/2018 12:49

Ring 5
✓ Succeeded
on 21/11/2018 12:57

1. Canary (internal users)
2. Smallest external data center
3. Largest external data center
4. International data centers
5. All the rest



Tony Thomas Yesterday 13:08
DeeDee queue TFS M158 to Rings 0-5

Collapse all

AzDeeDee Yesterday 13:08
Found AzureDevOps_M158_20190925.5

You want to queue AzureDevOps_M158_20190925.5 for TFS to rings 0 to 5

Is this correct?

Yes No

Tony Thomas Yesterday 13:08
AzDeeDee Yes

AzDeeDee Yesterday 13:08
Queueing release

Release queued

Reply

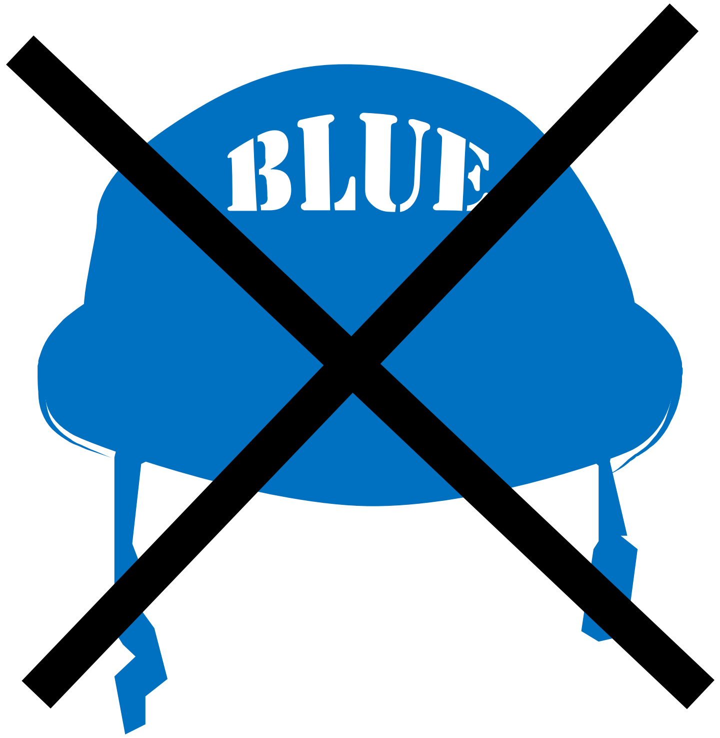
To: AzDeeDee

Type a new message

Security Mindset - Assume Breach

Started with war games to the learn attacks and practice response

VS.



- Initially double-blind test
- Over time, eliminated blue team

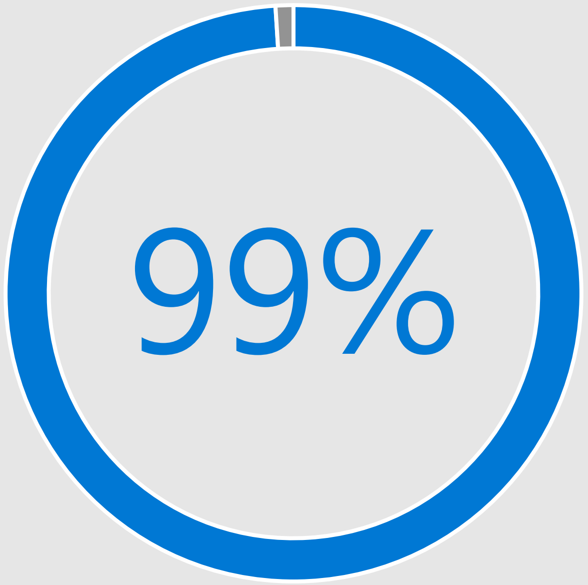
Our defenders need to be our defenders

Shifted left to prevent top risks

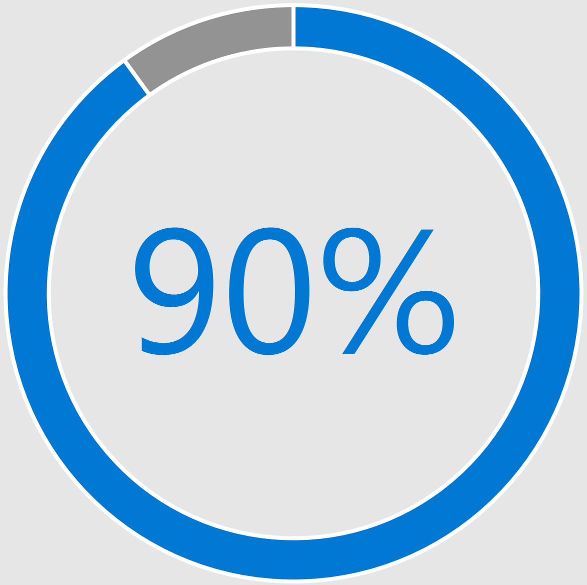
- Credential theft
- Secret leakage
- OSS vulnerabilities

Securing the Software Supply Chain

Leveraging open source fundamentally changes software development in the enterprise – manage the risks



of applications leverage open source software.



of new code bases are open source components.

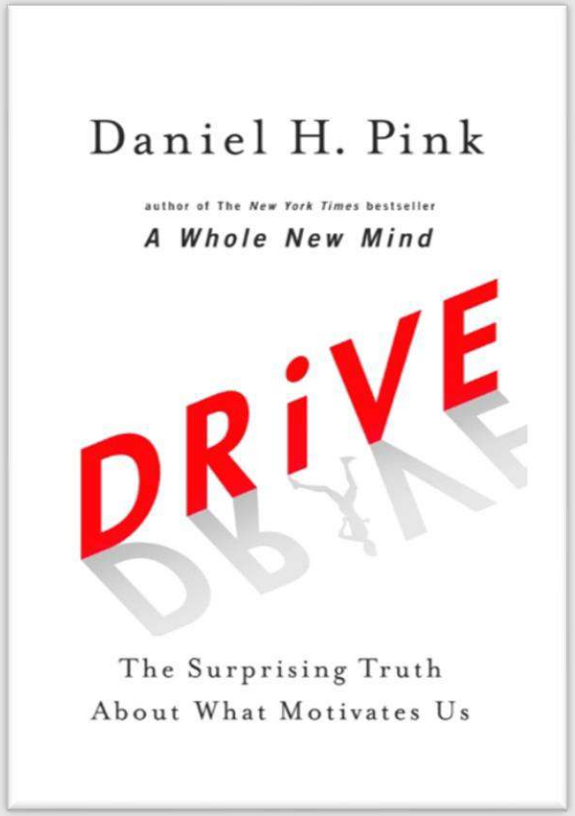
Source: Synopsys OSSRA 2019 report

Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

Agile at Scale with Aligned Autonomy

"Let's try to give our teams three things….
Autonomy, Mastery, Purpose"



- Plan
- Practices

Autonomy

- Organization
- Roles
- Teams
- Cadence
- Taxonomy

Alignment

Planning

Leadership is responsible for the big picture

Sprint
3 weeks
1

Quarter
4 sprints
4

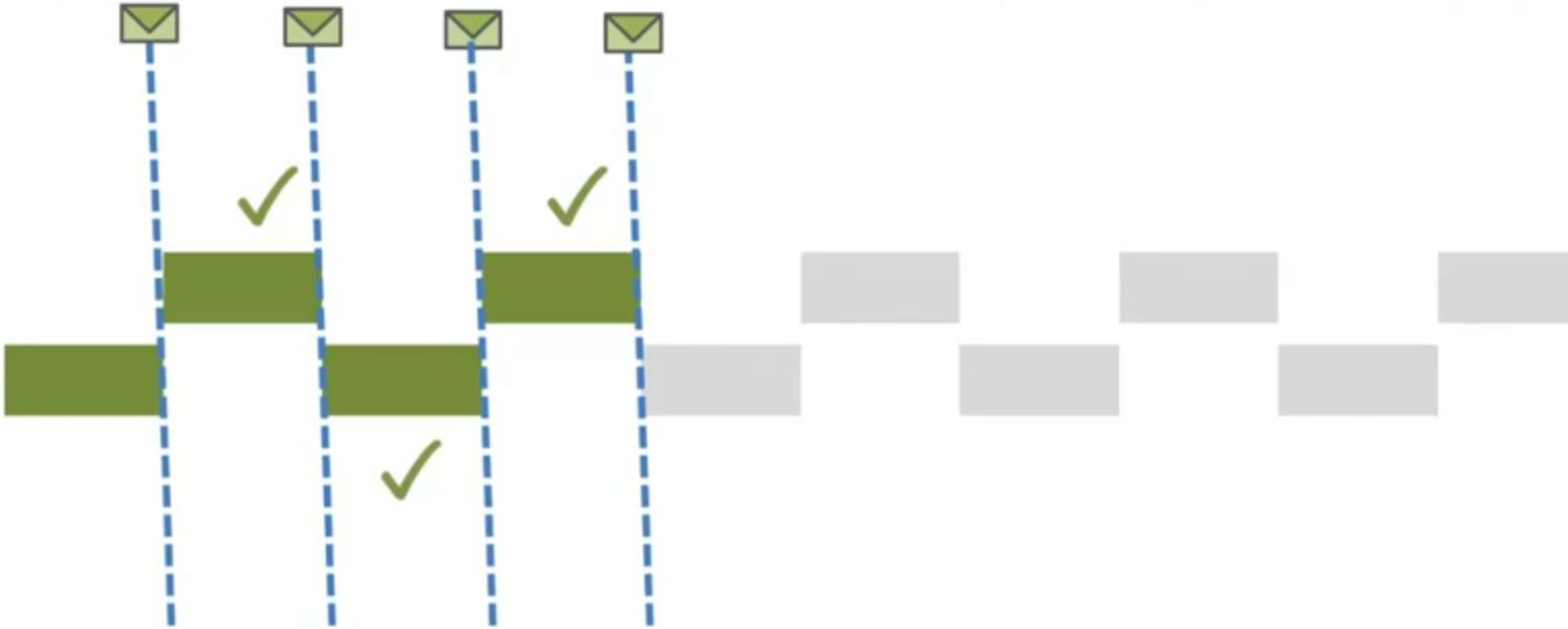
Semester
6 months
6

Strategy
12 months
12

Teams are responsible for the detail

Sprint mail

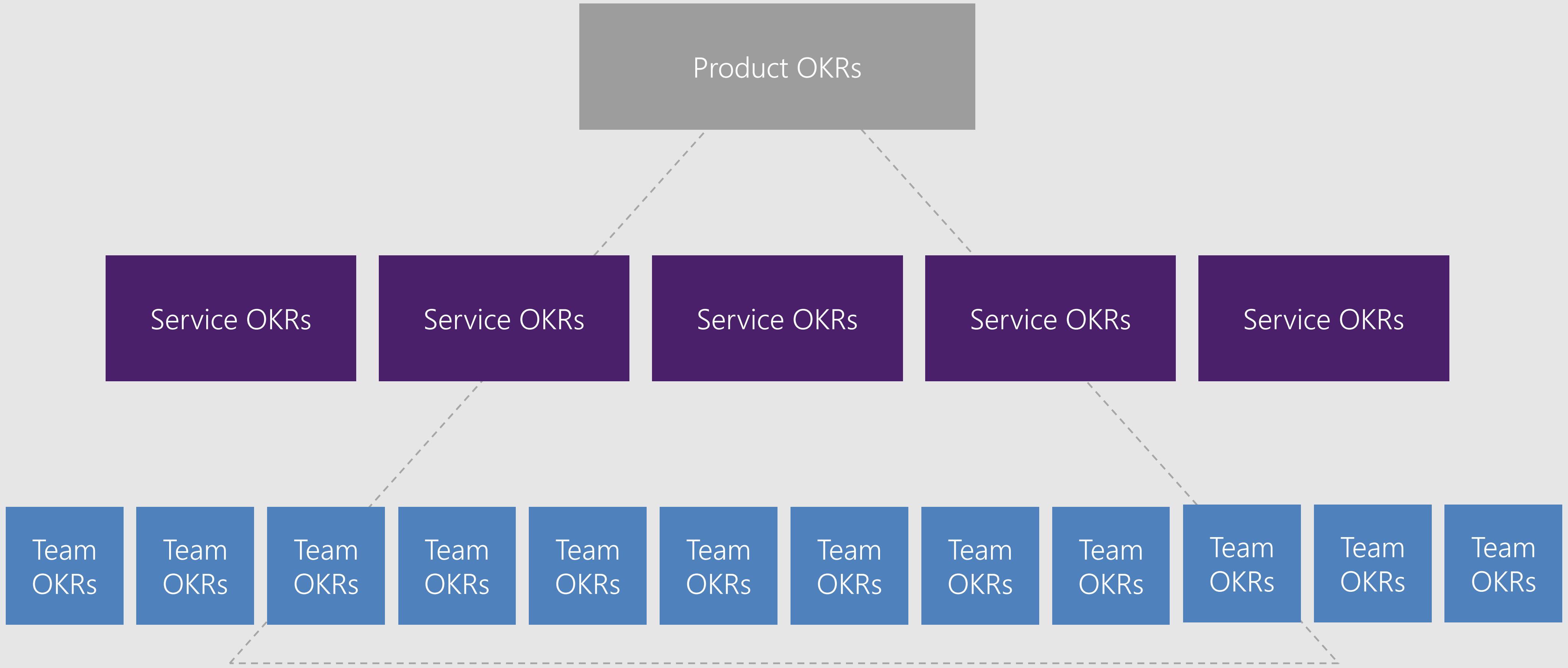
As needed: Experience Reviews



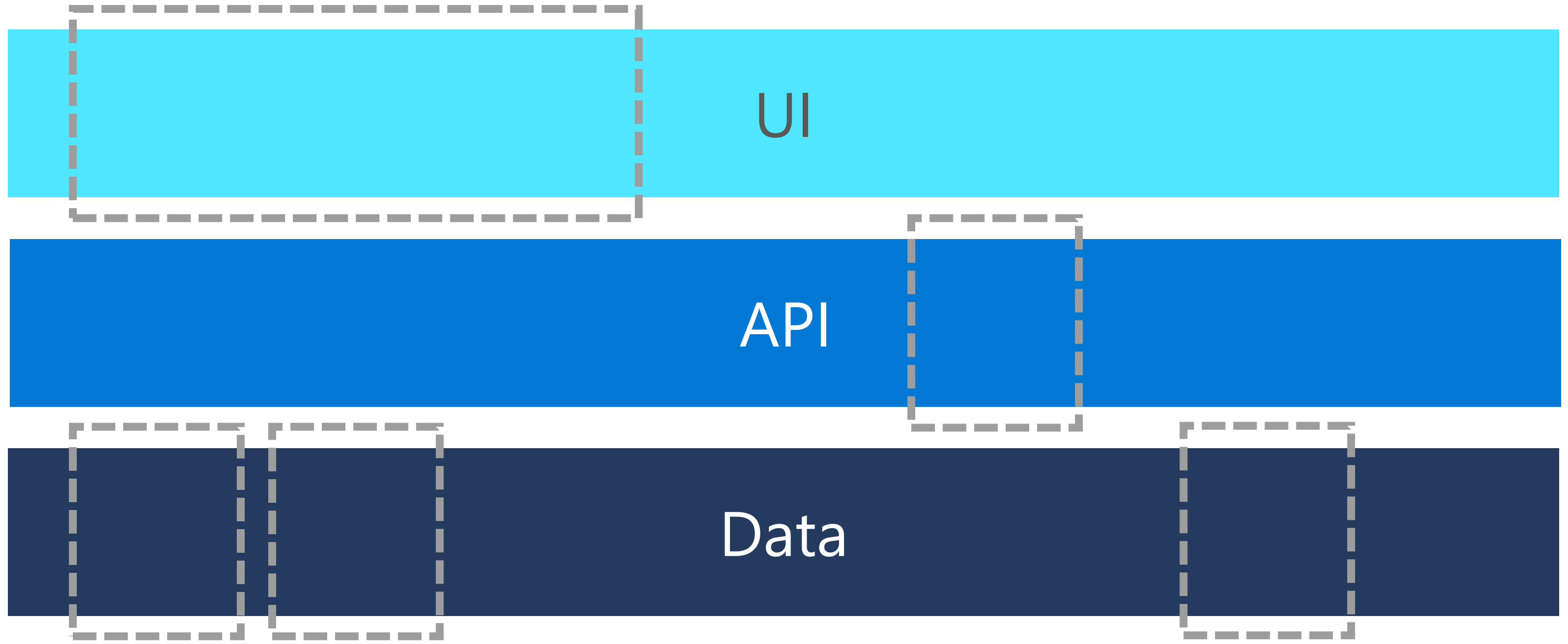
OKR check
2 sprints

OKR reset
4 sprints

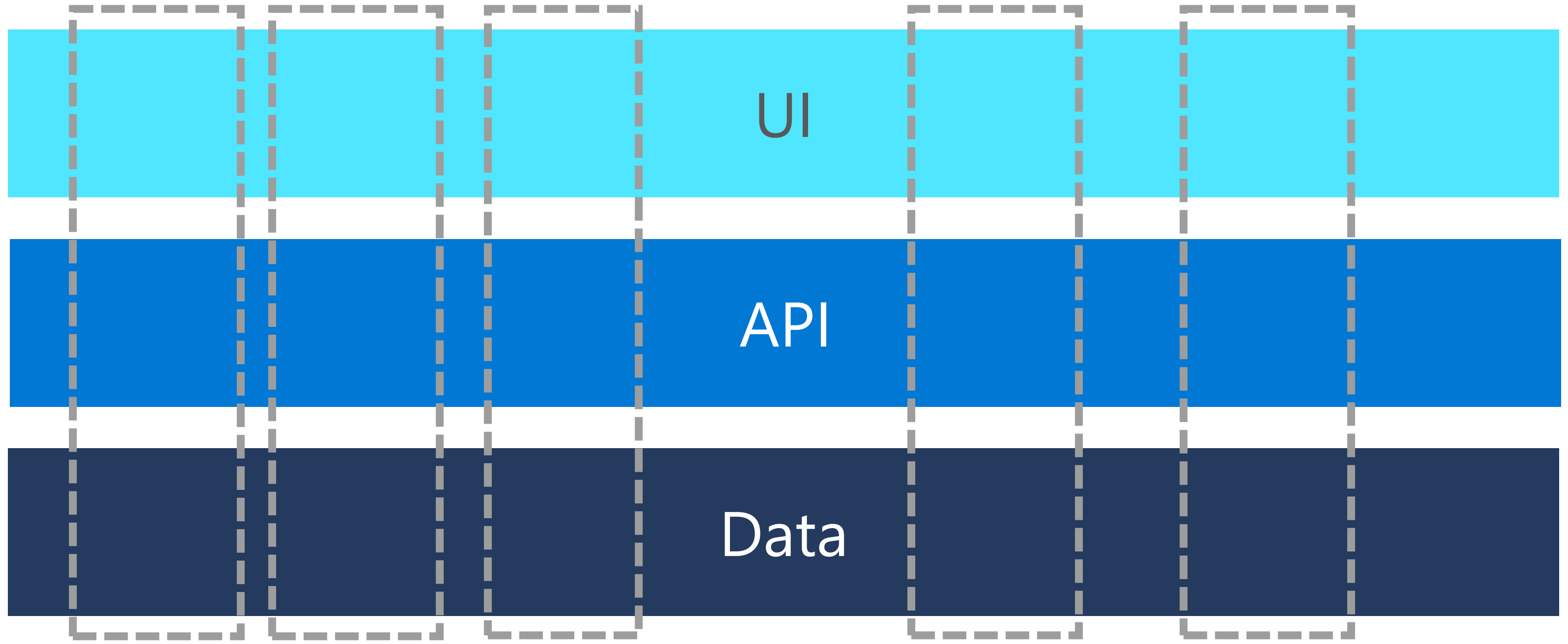
Alignment


Product OKRs
Service OKRs
Service OKRs
Service OKRs
Service OKRs
Service OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs
Team OKRs

Instead of Horizontal…



We strive for Vertical



Team Structure

ORG CHART

PROGRAM MANAGEMENT

DEVELOPMENT

TESTING

ORG CHART

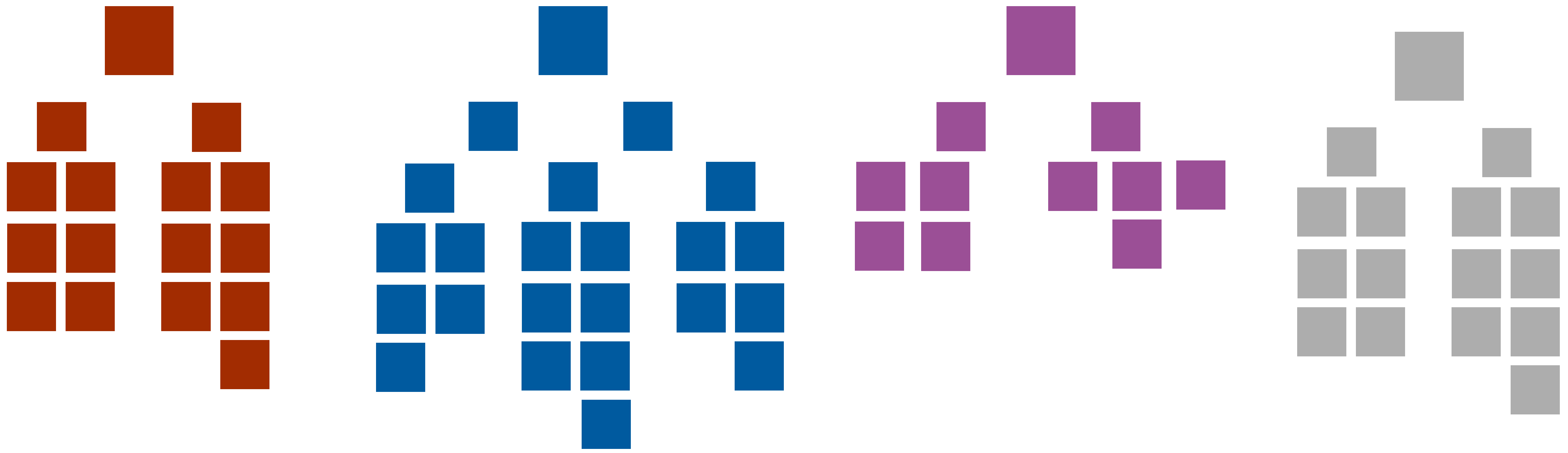
PROGRAM MANAGEMENT

ENGINEERING

OPs

Shift in roles and accountabilities – New way

Program | Engineering | Design | SRE



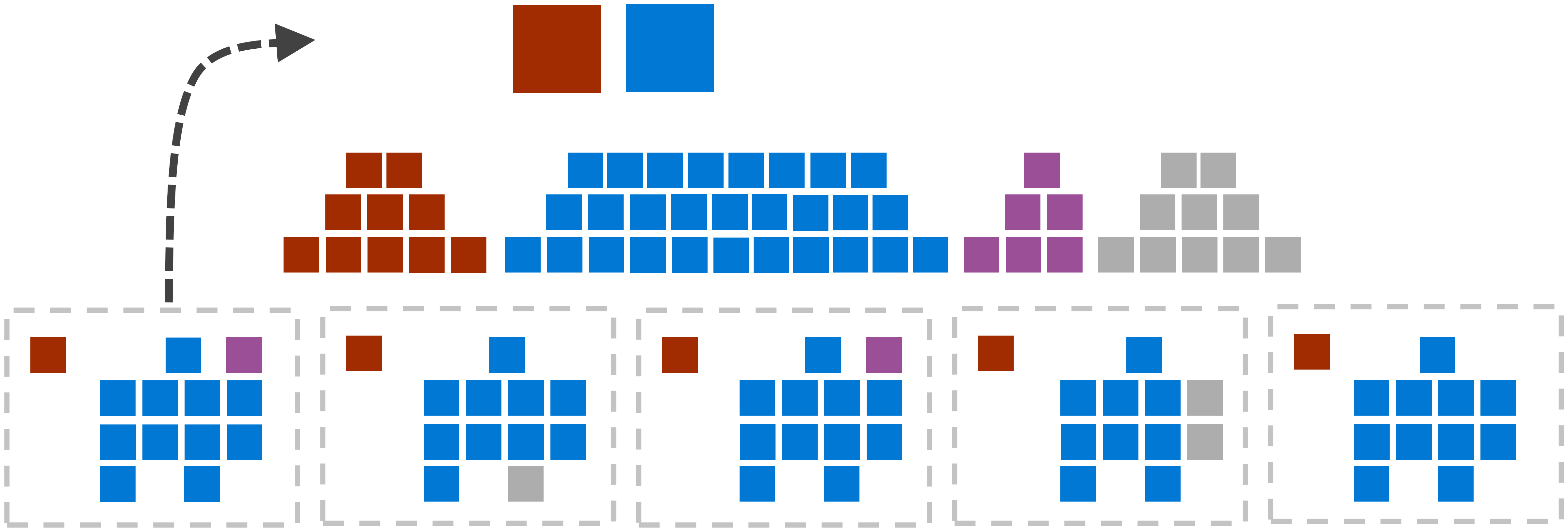
What we're building
Why we're building it

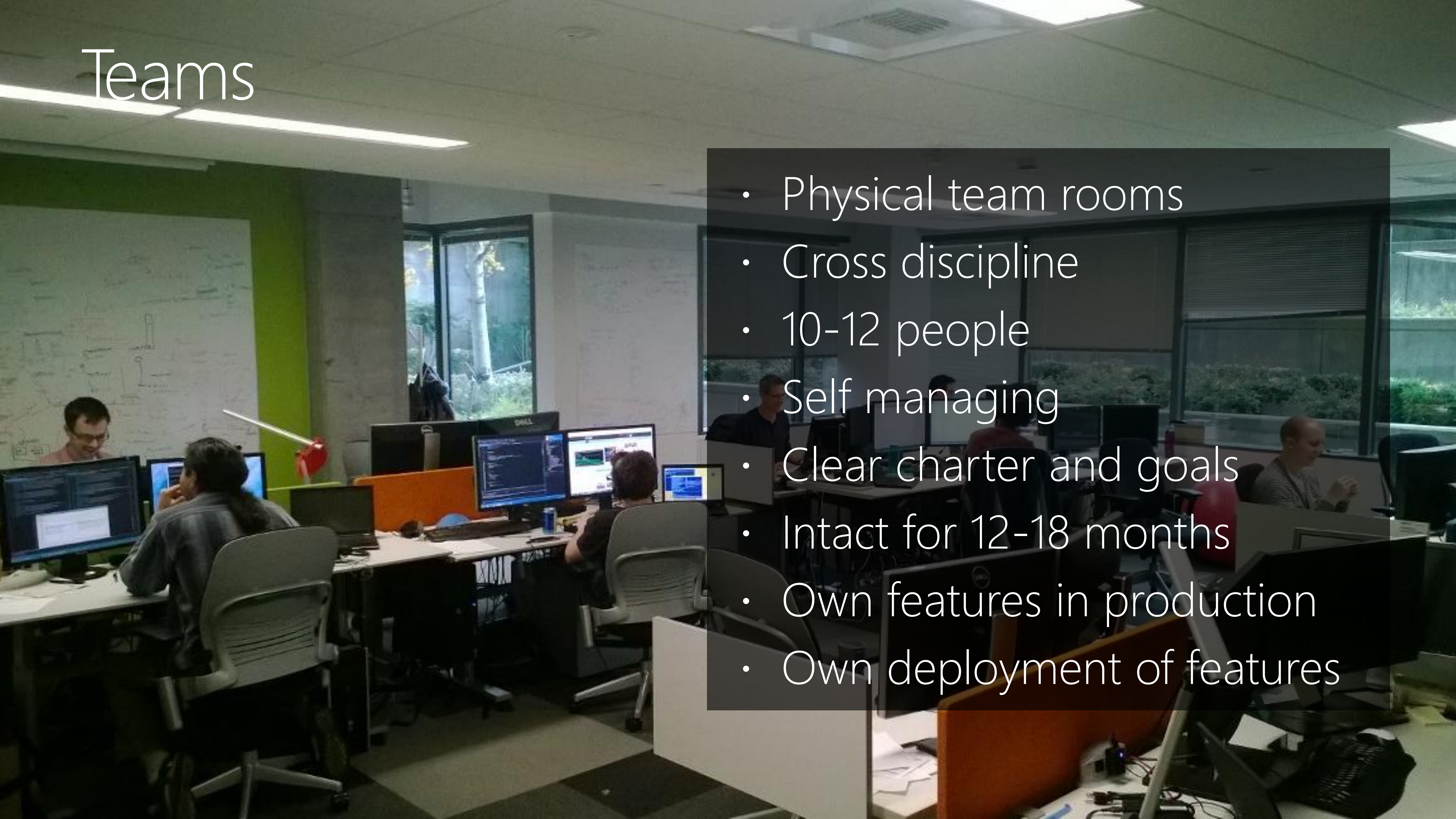
How we're building it
Building with Quality

User experience is
Pleasing and Usable

What we build is Reliable
and Available

Feature teams - Value focus





Teams

- Physical team rooms
- Cross discipline
- 10-12 people
- Self managing
- Clear charter and goals
- Intact for 12-18 months
- Own features in production
- Own deployment of features

Engineers Write Preferences for Teams

- Opportunity to change team without formality
- Employee choice, not manager driven

Transformation Benefits

- Teams feel that they own the customer experience & are responsible for improving it
- Teams are continually planning
- Planning is driven by continual learning
  - Telemetry on usage
  - Customer feedback
  - “Failing fast” through in incremental execution and delivery
- Opportunities to continually evaluate progress
- We can react… *if & when* we need to change course

Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

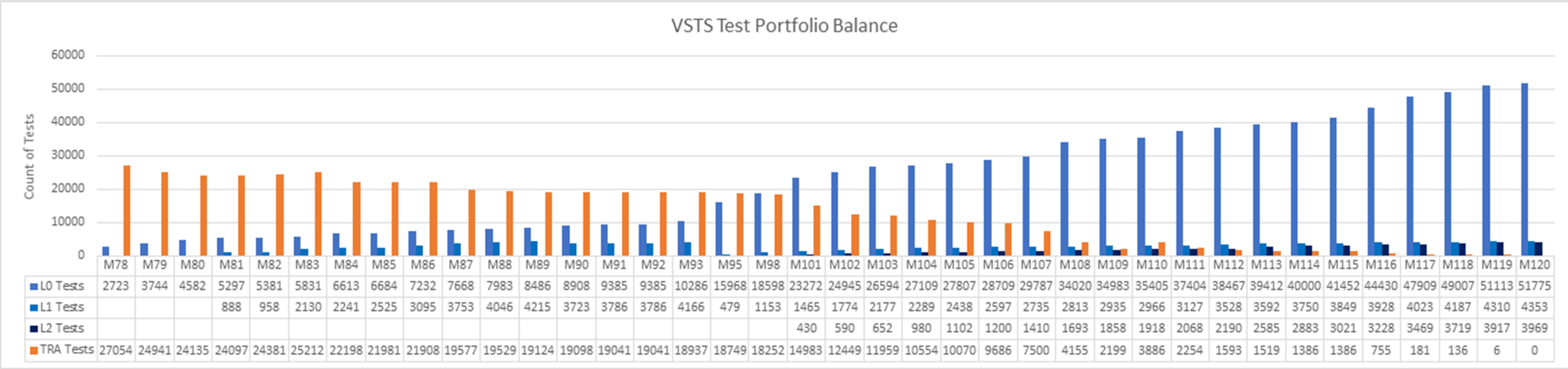
Testing: Shift Left from Integration to Unit

L0 – Requires only built binaries, no dependencies

L1 – Adds ability to use SQL and file system
Run L0 & L1 in the pull request builds

L2 – Test a service via REST APIs

L3 – Full environment to test end to end



	M78	M79	M80	M81	M82	M83	M84	M85	M86	M87	M88	M89	M90	M91	M92	M93	M95	M98
■ L0 Tests	2723	3744	4582	5297	5381	5831	6613	6684	7232	7668	7983	8486	8908	9385	9385	10286	15968	18598
■ L1 Tests				888	958	2130	2241	2525	3095	3753	4046	4215	3723	3786	3786	4166	479	1153
■ L2 Tests																		
■ TRA Tests	27054	24941	24135	24097	24381	25212	22198	21981	21908	19577	19529	19124	19098	19041	19041	18937	18749	18252

	M101	M102	M103	M104	M105	M106	M107	M108	M109	M110	M111	M112	M113	M114	M115	M116	M117	M118	M119	M120
■ L0 Tests	23272	24945	26594	27109	27807	28709	29787	34020	34983	35405	37404	38467	39412	40000	41452	44430	47909	49007	51113	51775
■ L1 Tests	1465	1774	2177	2289	2438	2597	2735	2813	2935	2966	3127	3528	3592	3750	3849	3928	4023	4187	4310	4353
■ L2 Tests	430	590	652	980	1102	1200	1410	1693	1858	1918	2068	2190	2585	2883	3021	3228	3469	3719	3917	3969
■ TRA Tests	14983	12449	11959	10554	10070	9686	7500	4155	2199	3886	2254	1593	1519	1386	1386	755	181	136	6	0

Pull Requests

PR's are point of code review

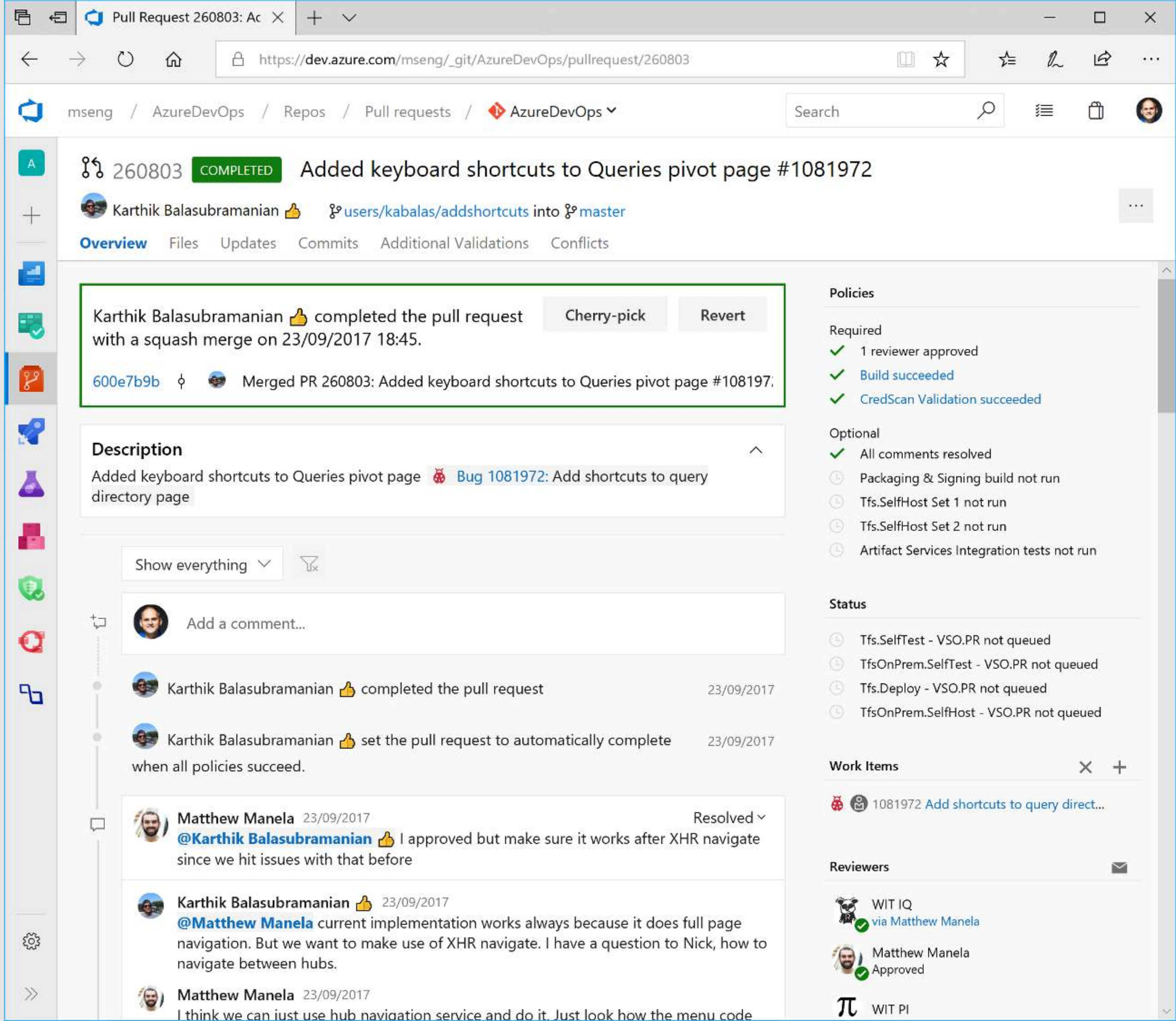
L0+L1 Tests performed before merge

Additional automated validation (compliance scanning etc)

Specific AD groups configured to require approval before merge

Result:

- Shift-left testing to pre-merge
- Makes CI build failures rare
- Accelerates the inner loop


Pull Request 260803: Ac

https://dev.azure.com/mseng/_git/AzureDevOps/pullrequest/260803

mseng / AzureDevOps / Repos / Pull requests / AzureDevOps

Search

260803 COMPLETED Added keyboard shortcuts to Queries pivot page #1081972

Karthik Balasubramanian users/kabalas/addshortcuts into master

Overview Files Updates Commits Additional Validations Conflicts

Karthik Balasubramanian completed the pull request with a squash merge on 23/09/2017 18:45.

Cherry-pick Revert

600e7b9b Merged PR 260803: Added keyboard shortcuts to Queries pivot page #108197

Description

Added keyboard shortcuts to Queries pivot page Bug 1081972: Add shortcuts to query directory page

Show everything

Add a comment...

Karthik Balasubramanian completed the pull request 23/09/2017

Karthik Balasubramanian set the pull request to automatically complete when all policies succeed. 23/09/2017

Matthew Manela 23/09/2017 Resolved
@Karthik Balasubramanian I approved but make sure it works after XHR navigate since we hit issues with that before

Karthik Balasubramanian 23/09/2017
@Matthew Manela current implementation works always because it does full page navigation. But we want to make use of XHR navigate. I have a question to Nick, how to navigate between hubs.

Matthew Manela 23/09/2017
I think we can just use hub navigation service and do it. Just look how the menu code

Policies

Required
- ✓ 1 reviewer approved
- ✓ Build succeeded
- ✓ CredScan Validation succeeded

Optional
- ✓ All comments resolved
- Packaging & Signing build not run
- Tfs.SelfHost Set 1 not run
- Tfs.SelfHost Set 2 not run
- Artifact Services Integration tests not run

Status
- Tfs.SelfTest - VSO.PR not queued
- TfsOnPrem.SelfTest - VSO.PR not queued
- Tfs.Deploy - VSO.PR not queued
- TfsOnPrem.SelfHost - VSO.PR not queued

Work Items

1081972 Add shortcuts to query direct...

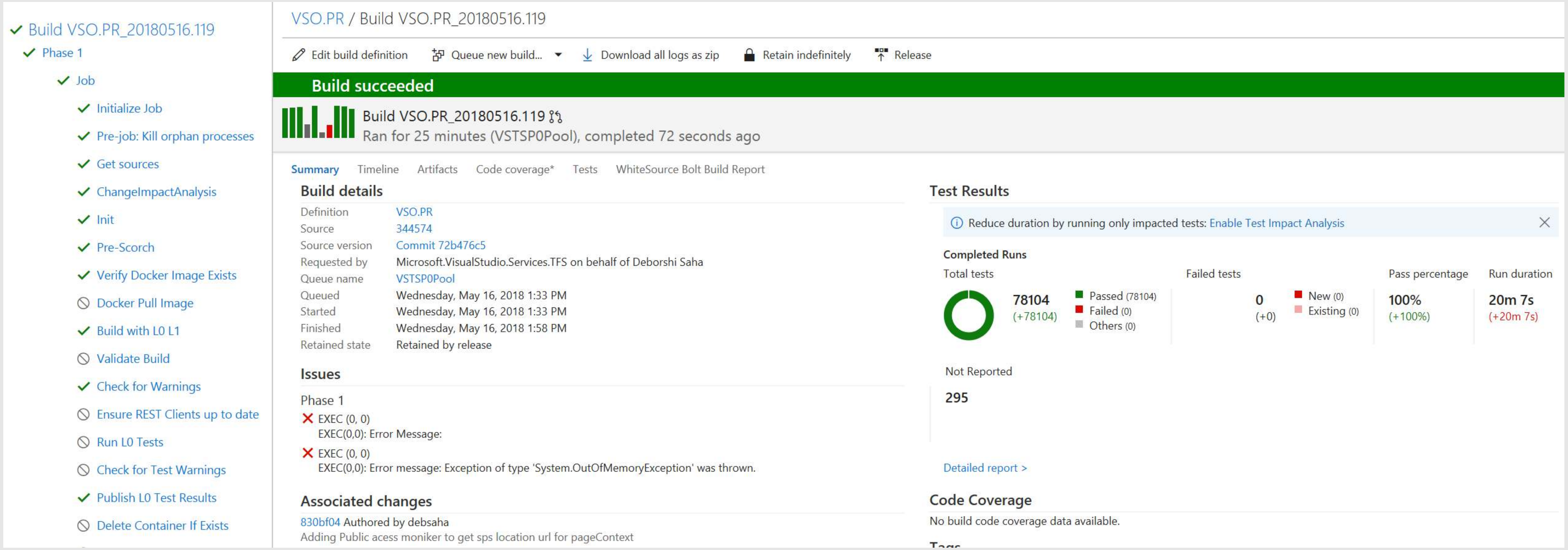
Reviewers

WIT IQ via Matthew Manela

Matthew Manela Approved

WIT PI

Tests Against the Pull Request



- ✓ Build VSO.PR_20180516.119
  - ✓ Phase 1
    - ✓ Job
      - ✓ Initialize Job
      - ✓ Pre-job: Kill orphan processes
      - ✓ Get sources
      - ✓ ChangeImpactAnalysis
      - ✓ Init
      - ✓ Pre-Scorch
      - ✓ Verify Docker Image Exists
      - ⦸ Docker Pull Image
      - ✓ Build with L0 L1
      - ⦸ Validate Build
      - ✓ Check for Warnings
      - ⦸ Ensure REST Clients up to date
      - ⦸ Run L0 Tests
      - ⦸ Check for Test Warnings
      - ✓ Publish L0 Test Results
      - ⦸ Delete Container If Exists

VSO.PR / Build VSO.PR_20180516.119

Build succeeded

Build VSO.PR_20180516.119
Ran for 25 minutes (VSTSP0Pool), completed 72 seconds ago

Build details

Definition	VSO.PR
Source	344574
Source version	Commit 72b476c5
Requested by	Microsoft.VisualStudio.Services.TFS on behalf of Deborshi Saha
Queue name	VSTSP0Pool
Queued	Wednesday, May 16, 2018 1:33 PM
Started	Wednesday, May 16, 2018 1:33 PM
Finished	Wednesday, May 16, 2018 1:58 PM
Retained state	Retained by release

Issues

Phase 1
✗ EXEC (0, 0)
EXEC(0,0): Error Message:
✗ EXEC (0, 0)
EXEC(0,0): Error message: Exception of type 'System.OutOfMemoryException' was thrown.

Associated changes

830bf04 Authored by debsaha
Adding Public acess moniker to get sps location url for pageContext

Test Results

ⓘ Reduce duration by running only impacted tests: Enable Test Impact Analysis

Completed Runs

Total tests		Failed tests		Pass percentage	Run duration
78104 (+78104)	🟩 Passed (78104) 🟥 Failed (0) ⬜ Others (0)	0 (+0)	🟥 New (0) 🟥 Existing (0)	100% (+100%)	20m 7s (+20m 7s)

Not Reported
295

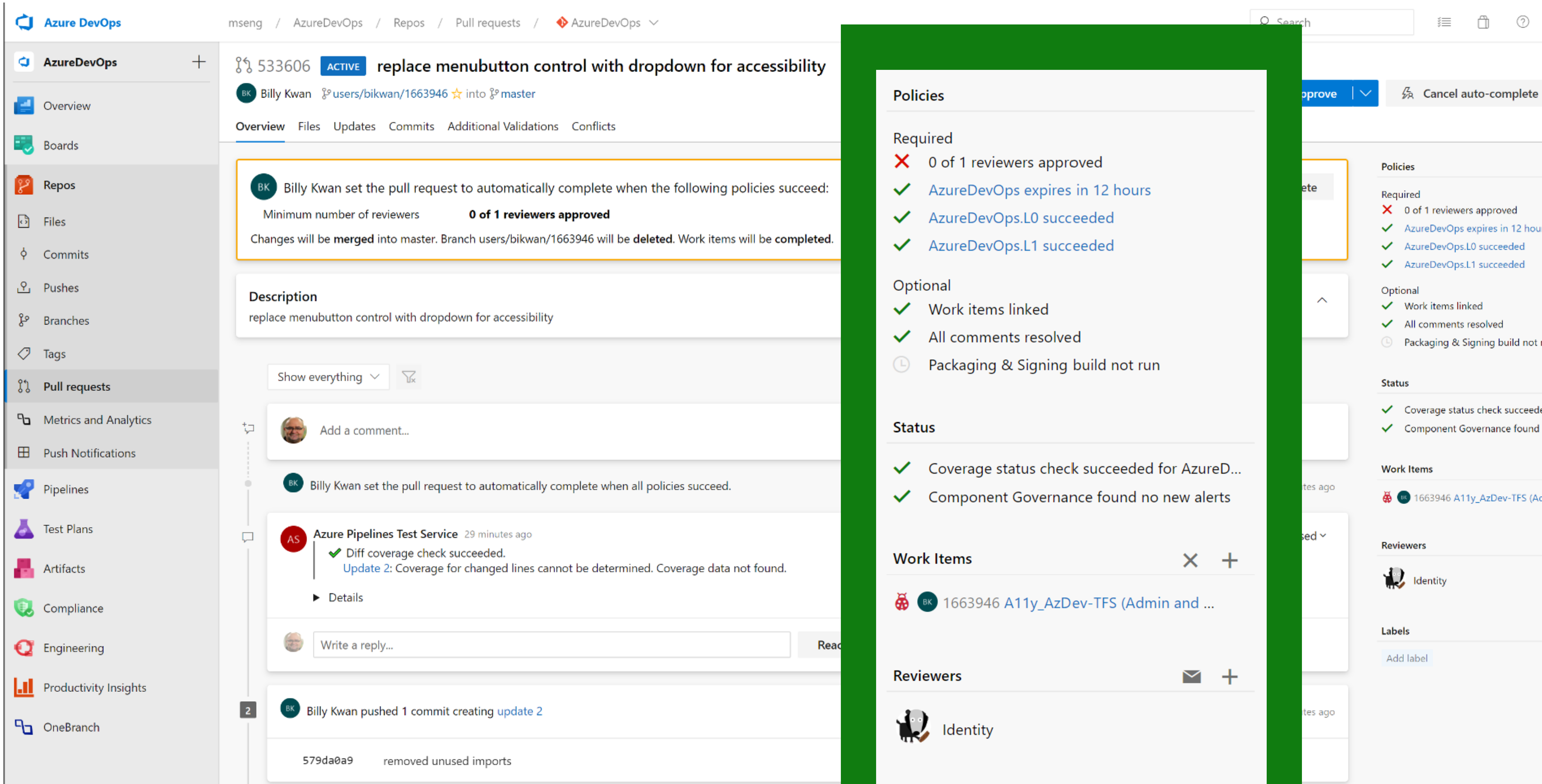
Detailed report >

Code Coverage

No build code coverage data available.

Feedback in minutes, before acceptance of PR

Pull Requests Control Code Merge to Master



533606 ACTIVE replace menubutton control with dropdown for accessibility

Billy Kwan users/bikwan/1663946 into master

Billy Kwan set the pull request to automatically complete when the following policies succeed:

Minimum number of reviewers **0 of 1 reviewers approved**

Changes will be **merged** into master. Branch users/bikwan/1663946 will be **deleted**. Work items will be **completed**.

Description

replace menubutton control with dropdown for accessibility

Billy Kwan set the pull request to automatically complete when all policies succeed.

Azure Pipelines Test Service 29 minutes ago

✓ Diff coverage check succeeded.

Update 2: Coverage for changed lines cannot be determined. Coverage data not found.

2 Billy Kwan pushed 1 commit creating update 2

579da0a9 removed unused imports

Policies

Required

✗ 0 of 1 reviewers approved

✓ AzureDevOps expires in 12 hours

✓ AzureDevOps.L0 succeeded

✓ AzureDevOps.L1 succeeded

Optional

✓ Work items linked

✓ All comments resolved

Packaging & Signing build not run

Status

✓ Coverage status check succeeded for AzureD...

✓ Component Governance found no new alerts

Work Items

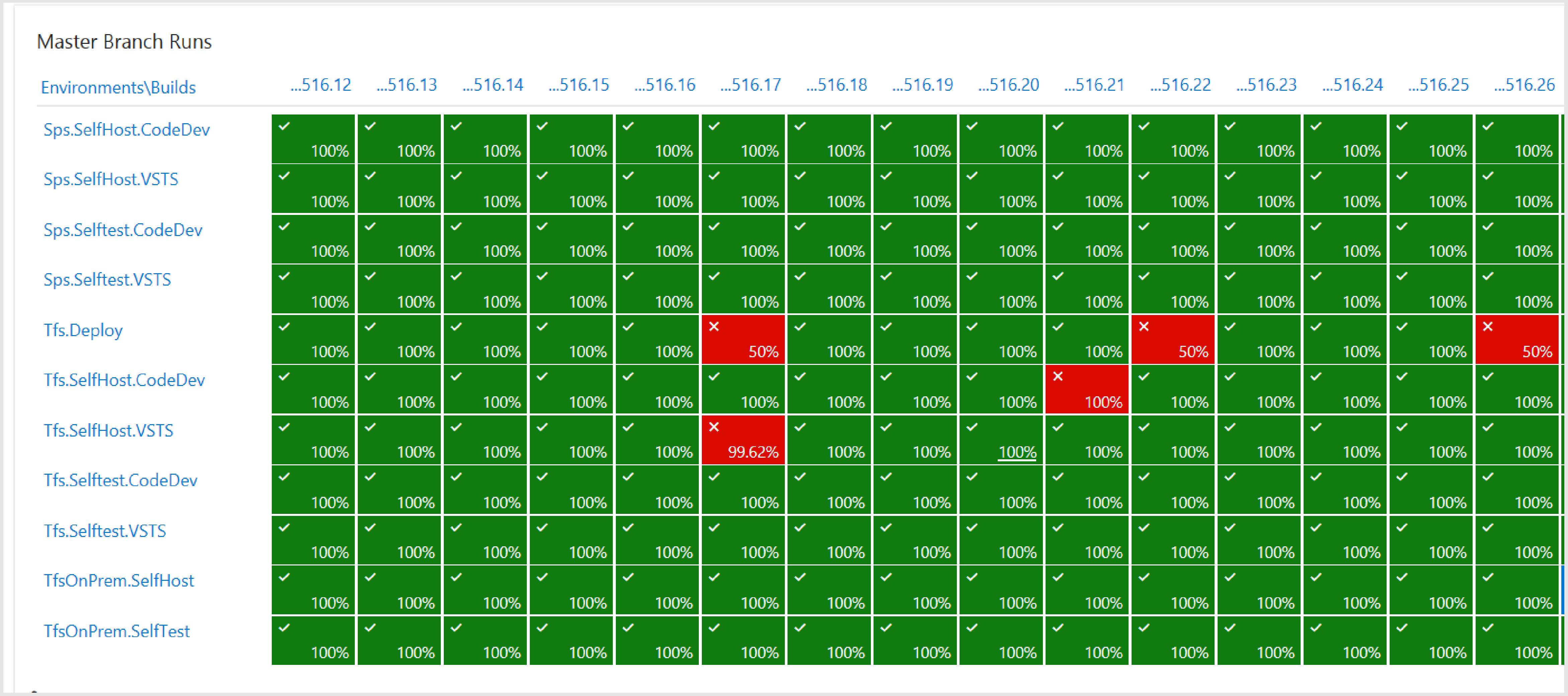
1663946 A11y_AzDev-TFS (Admin and ...

Reviewers

Identity

Policies
Required
✗ 0 of 1 reviewers approved
✓ AzureDevOps expires in 12 hou
✓ AzureDevOps.L0 succeeded
✓ AzureDevOps.L1 succeeded
Optional
✓ Work items linked
✓ All comments resolved
Packaging & Signing build not
Status
✓ Coverage status check succeede
✓ Component Governance found
Work Items
1663946 A11y_AzDev-TFS (Ac
Reviewers
Identity
Labels
Add label

Green Means Green, Red Means Red



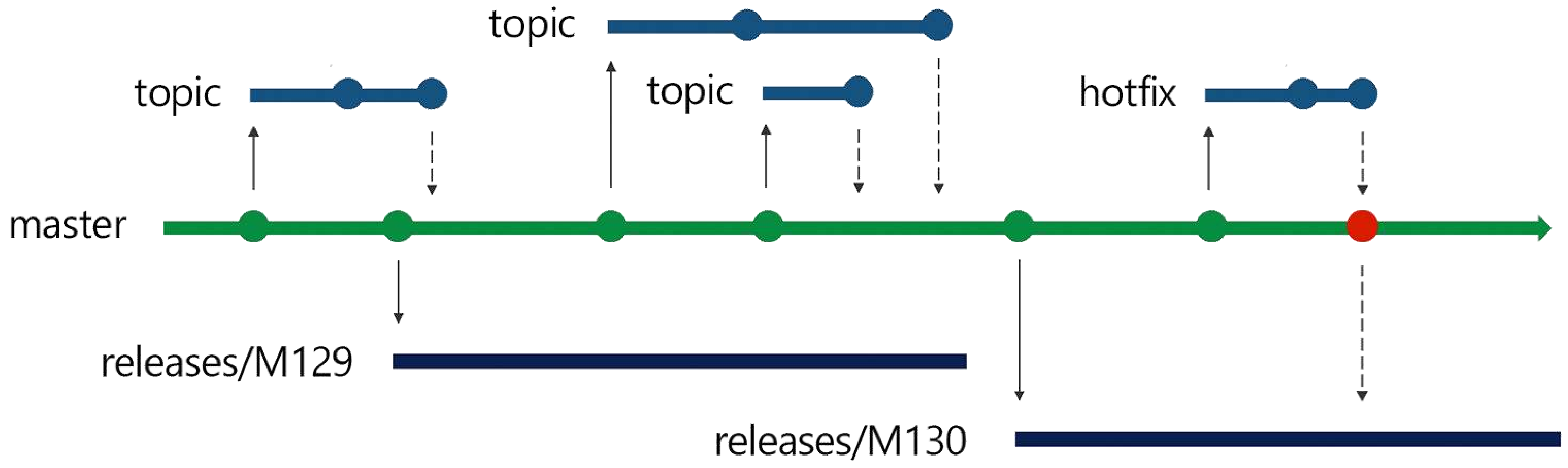
Master Branch Runs

Environments\Builds	...516.12	...516.13	...516.14	...516.15	...516.16	...516.17	...516.18	...516.19	...516.20	...516.21	...516.22	...516.23	...516.24	...516.25	...516.26
Sps.SelfHost.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.SelfHost.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.Selftest.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.Selftest.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	× 50%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	× 50%	✓ 100%	✓ 100%	✓ 100%	× 50%
Tfs.SelfHost.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	× 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.SelfHost.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	× 99.62%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Selftest.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.Selftest.VSTS	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%

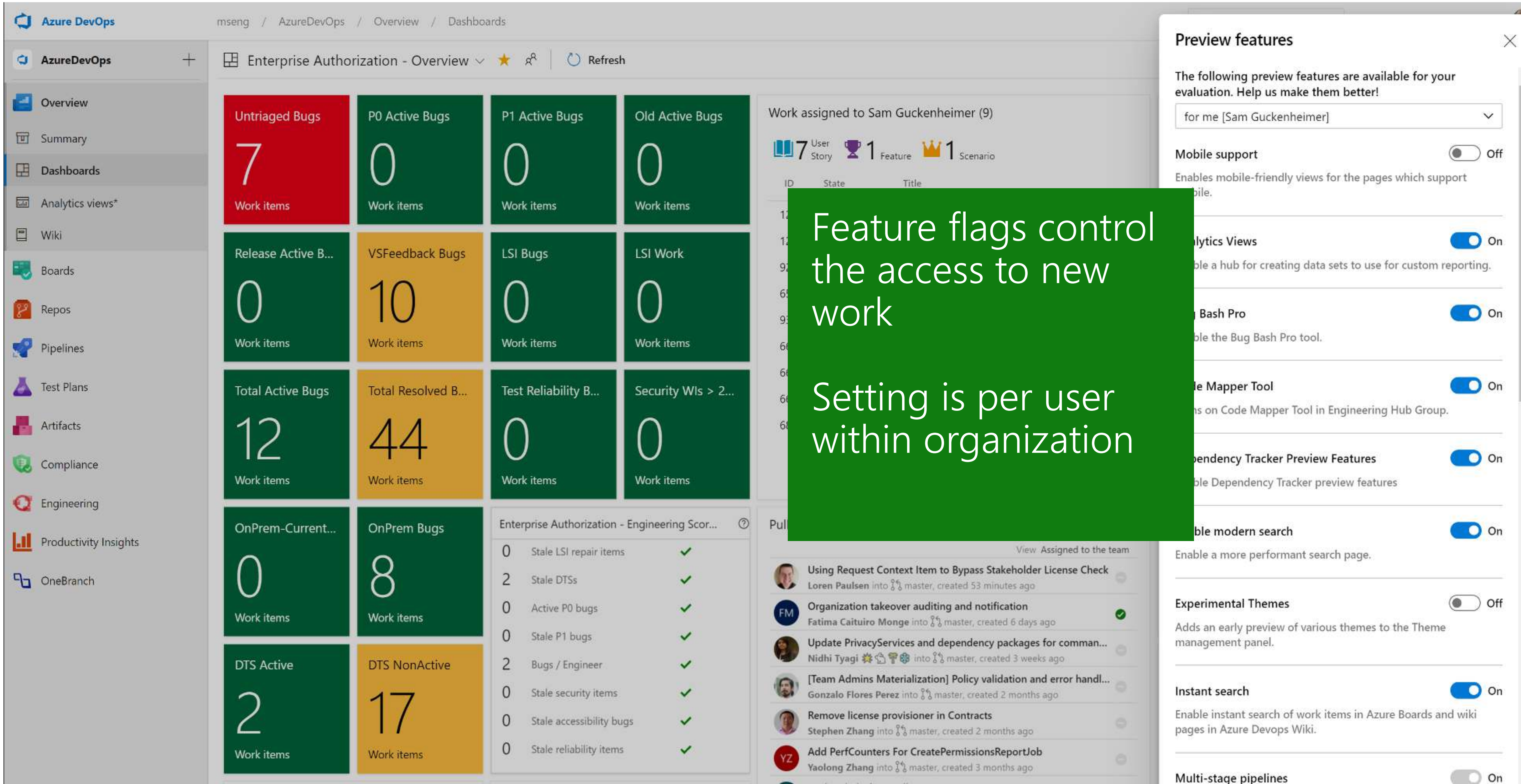
Only all-green builds get to release

Branching Structure

Using Trunk Based Development to avoid Merge Debt



Progressive experimentation



Azure DevOps   mseng / AzureDevOps / Overview / Dashboards

AzureDevOps

- Overview
- Summary
- Dashboards
- Analytics views*
- Wiki
- Boards
- Repos
- Pipelines
- Test Plans
- Artifacts
- Compliance
- Engineering
- Productivity Insights
- OneBranch

Enterprise Authorization - Overview   Refresh

Untriaged Bugs 7 Work items	P0 Active Bugs 0 Work items	P1 Active Bugs 0 Work items	Old Active Bugs 0 Work items
Release Active B... 0 Work items	VSFeedback Bugs 10 Work items	LSI Bugs 0 Work items	LSI Work 0 Work items
Total Active Bugs 12 Work items	Total Resolved B... 44 Work items	Test Reliability B... 0 Work items	Security WIs > 2... 0 Work items
OnPrem-Current... 0 Work items	OnPrem Bugs 8 Work items		
DTS Active 2 Work items	DTS NonActive 17 Work items		

Work assigned to Sam Guckenheimer (9)

7 User Story   1 Feature   1 Scenario

ID   State   Title

Enterprise Authorization - Engineering Scor...

0	Stale LSI repair items	✓
2	Stale DTSs	✓
0	Active P0 bugs	✓
0	Stale P1 bugs	✓
2	Bugs / Engineer	✓
0	Stale security items	✓
0	Stale accessibility bugs	✓
0	Stale reliability items	✓

Pul

View   Assigned to the team

- Using Request Context Item to Bypass Stakeholder License Check — Loren Paulsen into master, created 53 minutes ago
- Organization takeover auditing and notification — Fatima Caituiro Monge into master, created 6 days ago
- Update PrivacyServices and dependency packages for comman... — Nidhi Tyagi into master, created 3 weeks ago
- [Team Admins Materialization] Policy validation and error handl... — Gonzalo Flores Perez into master, created 2 months ago
- Remove license provisioner in Contracts — Stephen Zhang into master, created 2 months ago
- Add PerfCounters For CreatePermissionsReportJob — Yaolong Zhang into master, created 3 months ago

Preview features

The following preview features are available for your evaluation. Help us make them better!

for me [Sam Guckenheimer]

Mobile support — Enables mobile-friendly views for the pages which support ...bile.	Off
...lytics Views — ...ble a hub for creating data sets to use for custom reporting.	On
... Bash Pro — ...ble the Bug Bash Pro tool.	On
...le Mapper Tool — ...s on Code Mapper Tool in Engineering Hub Group.	On
...endency Tracker Preview Features — ...ble Dependency Tracker preview features	On
...ble modern search — Enable a more performant search page.	On
Experimental Themes — Adds an early preview of various themes to the Theme management panel.	Off
Instant search — Enable instant search of work items in Azure Boards and wiki pages in Azure Devops Wiki.	On
Multi-stage pipelines	On

Feature flags control the access to new work

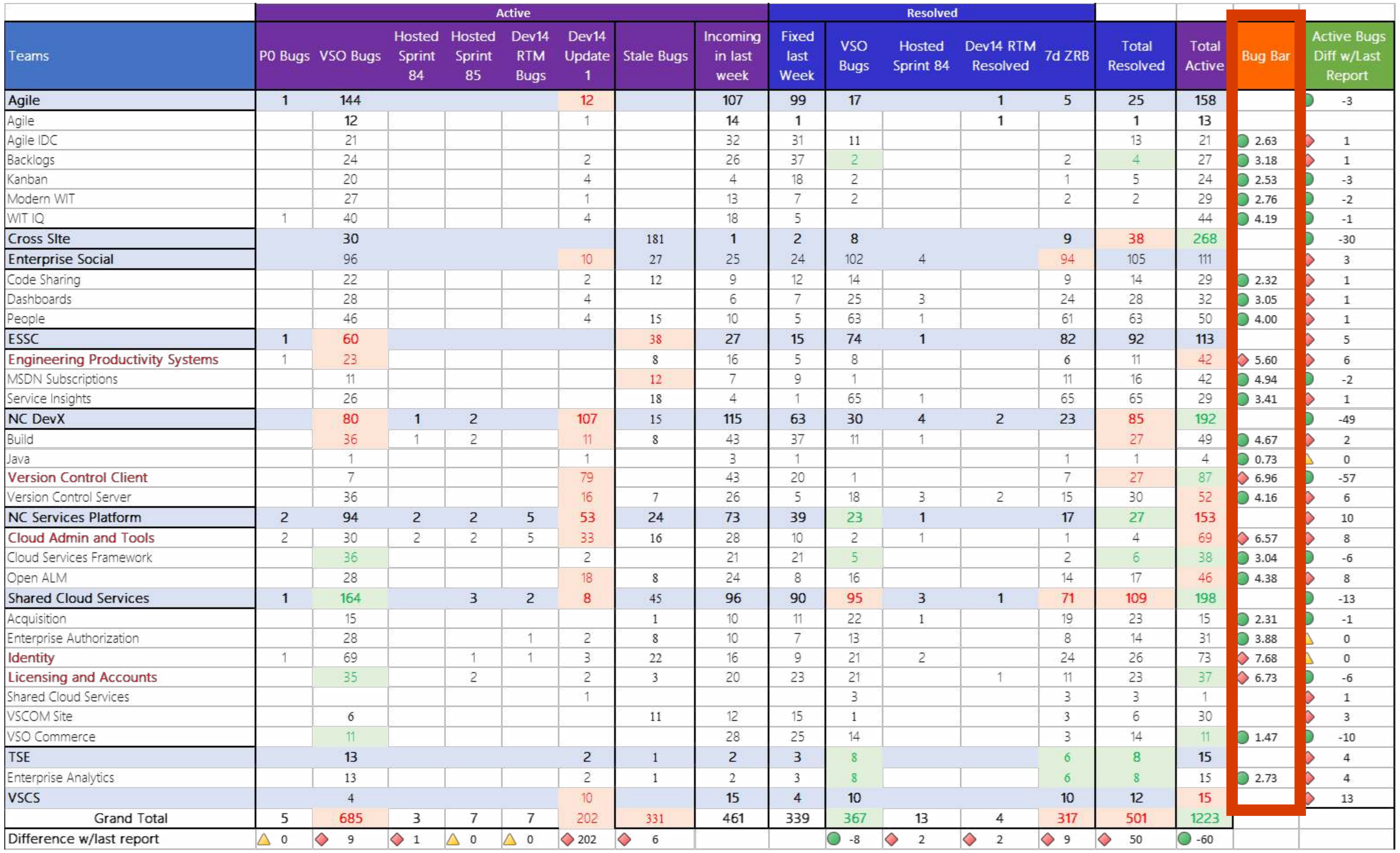
Setting is per user within organization

Do Not Incur Debt

We all follow a simple rule we call the "Bug Cap":

# engineers on your team x 4 = ?

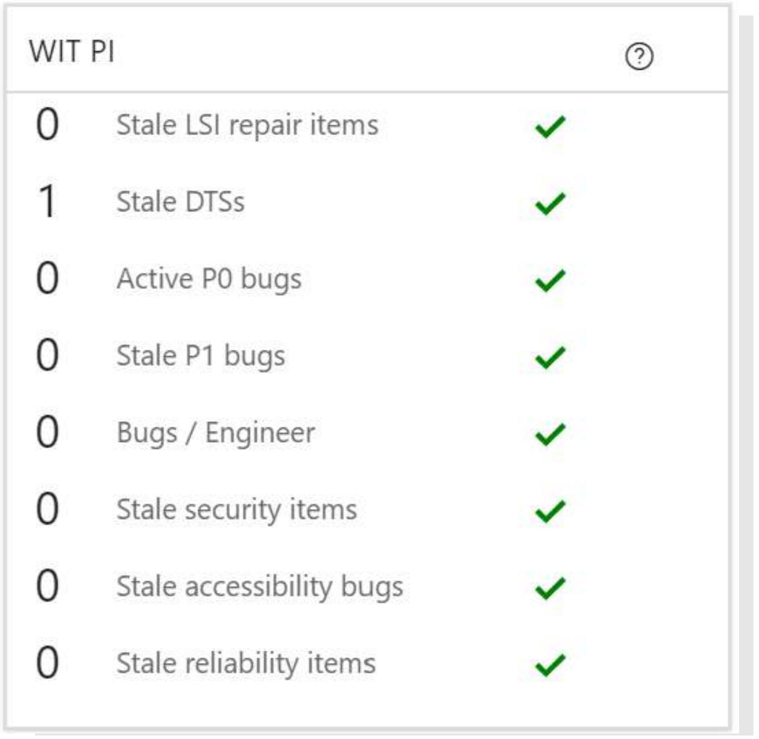
Bugs



	Active								Resolved								
Teams	P0 Bugs	VSO Bugs	Hosted Sprint 84	Hosted Sprint 85	Dev14 RTM Bugs	Dev14 Update 1	Stale Bugs	Incoming in last week	Fixed last Week	VSO Bugs	Hosted Sprint 84	Dev14 RTM Resolved	7d ZRB	Total Resolved	Total Active	Bug Bar	Active Bugs Diff w/Last Report
Agile	1	144				12		107	99	17		1	5	25	158		● -3
Agile		12				1		14	1			1		1	13		
Agile IDC		21						32	31	11				13	21	● 2.63	◆ 1
Backlogs		24				2		26	37	2			2	4	27	● 3.18	◆ 1
Kanban		20				4		4	18	2			1	5	24	● 2.53	● -3
Modern WIT		27				1		13	7	2			2	2	29	● 2.76	● -2
WIT IQ	1	40				4		18	5						44	● 4.19	● -1
Cross SIte		30					181	1	2	8			9	38	268		● -30
Enterprise Social		96				10	27	25	24	102	4		94	105	111		◆ 3
Code Sharing		22				2	12	9	12	14			9	14	29	● 2.32	◆ 1
Dashboards		28				4		6	7	25	3		24	28	32	● 3.05	◆ 1
People		46				4	15	10	5	63	1		61	63	50	● 4.00	◆ 1
ESSC	1	60					38	27	15	74	1		82	92	113		◆ 5
Engineering Productivity Systems	1	23					8	16	5	8			6	11	42	◆ 5.60	◆ 6
MSDN Subscriptions		11					12	7	9	1			11	16	42	● 4.94	● -2
Service Insights		26					18	4	1	65	1		65	65	29	● 3.41	◆ 1
NC DevX		80	1	2		107	15	115	63	30	4	2	23	85	192		● -49
Build		36	1	2		11	8	43	37	11	1			27	49	● 4.67	◆ 2
Java		1				1		3	1				1	1	4	● 0.73	▲ 0
Version Control Client		7				79		43	20	1			7	27	87	◆ 6.96	● -57
Version Control Server		36				16	7	26	5	18	3	2	15	30	52	● 4.16	◆ 6
NC Services Platform	2	94	2	2	5	53	24	73	39	23	1		17	27	153		◆ 10
Cloud Admin and Tools	2	30	2	2	5	33	16	28	10	2	1		1	4	69	◆ 6.57	◆ 8
Cloud Services Framework		36				2		21	21	5			2	6	38	● 3.04	● -6
Open ALM		28				18	8	24	8	16			14	17	46	● 4.38	◆ 8
Shared Cloud Services	1	164		3	2	8	45	96	90	95	3	1	71	109	198		● -13
Acquisition		15					1	10	11	22	1		19	23	15	● 2.31	● -1
Enterprise Authorization		28			1	2	8	10	7	13			8	14	31	● 3.88	▲ 0
Identity	1	69		1	1	3	22	16	9	21	2		24	26	73	◆ 7.68	▲ 0
Licensing and Accounts		35		2		2	3	20	23	21		1	11	23	37	◆ 6.73	● -6
Shared Cloud Services						1				3			3	3	1		◆ 1
VSCOM Site		6					11	12	15	1			3	6	30		◆ 3
VSO Commerce		11						28	25	14			3	14	11	● 1.47	● -10
TSE		13				2	1	2	3	8			6	8	15		◆ 4
Enterprise Analytics		13				2	1	2	3	8			6	8	15	● 2.73	◆ 4
VSCS		4				10		15	4	10			10	12	15		◆ 13
Grand Total	5	685	3	7	7	202	331	461	339	367	13	4	317	501	1223		
Difference w/last report	▲ 0	◆ 9	◆ 1	▲ 0	▲ 0	◆ 202	◆ 6			● -8	◆ 2	◆ 2	◆ 9	◆ 50	● -60		

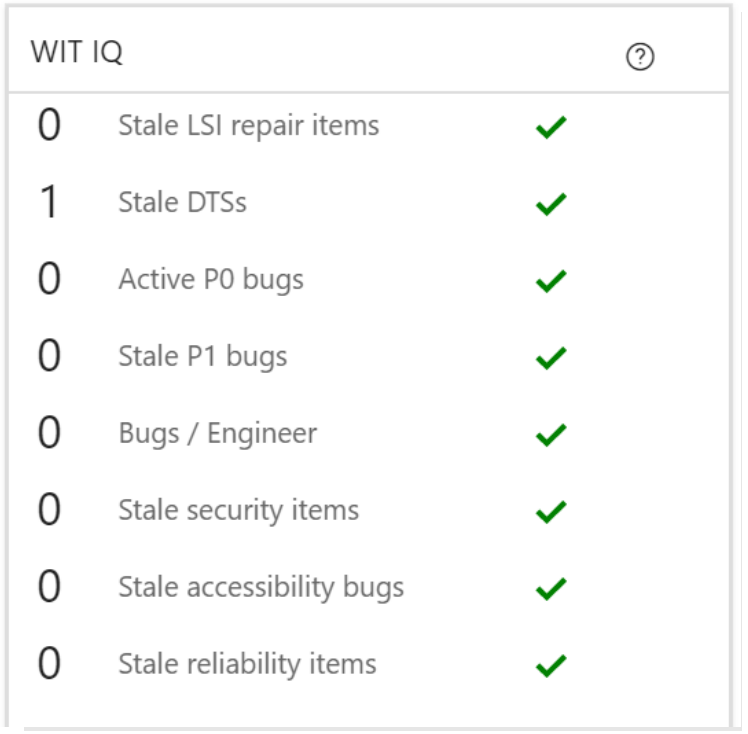
Stay Clean

Make technical debt visible on every team’s dashboard



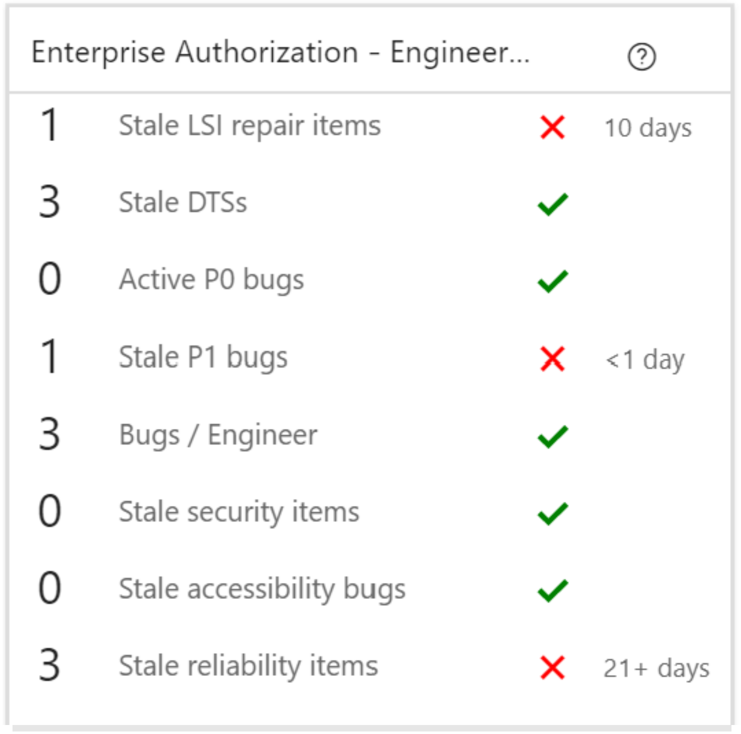
WIT PI ⓘ

0	Stale LSI repair items	✓
1	Stale DTSs	✓
0	Active P0 bugs	✓
0	Stale P1 bugs	✓
0	Bugs / Engineer	✓
0	Stale security items	✓
0	Stale accessibility bugs	✓
0	Stale reliability items	✓



WIT IQ ⓘ

0	Stale LSI repair items	✓
1	Stale DTSs	✓
0	Active P0 bugs	✓
0	Stale P1 bugs	✓
0	Bugs / Engineer	✓
0	Stale security items	✓
0	Stale accessibility bugs	✓
0	Stale reliability items	✓



Enterprise Authorization - Engineer... ⓘ

1	Stale LSI repair items	✗	10 days
3	Stale DTSs	✓	
0	Active P0 bugs	✓	
1	Stale P1 bugs	✗	<1 day
3	Bugs / Engineer	✓	
0	Stale security items	✓	
0	Stale accessibility bugs	✓	
3	Stale reliability items	✗	21+ days

Shifting Right: Service Health Review

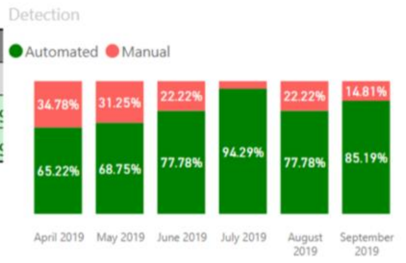
Health: Service Availability & Health Metrics

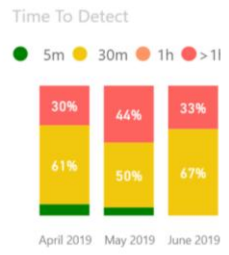
Note: will drop in Sept Availability on Monday when available

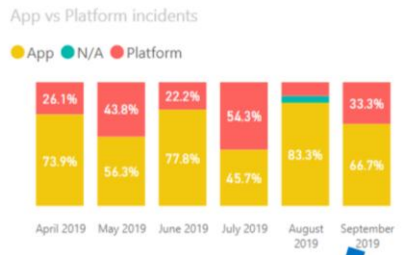
Azure DevOps	Target	Apr	May	Jun	Jul	Aug	Sep	Sep
Service Health - Availability								
TFS	99.90%	99.95	99.94	99.93	99.82	99.96	99.96	99.9
SPS	99.99%	99.996	99.998	99.995	99.993	99.996	99.996	99.9

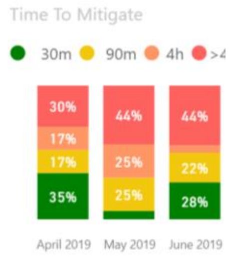
Service	Apr	May	Jun	Jul	Aug	Sep	% TTM < 90Min
ARTIFACT	1	1	1		1	1	
Blobstore						2	50%
Packaging				3		1	
Pipelines	2		2	1	4	5	80%
SPS	3	1	5	4		2	100%
TCM	1		3	1			
TFS	11	6	7	18	8	11	82%
User	1	1		4		1	
OTHER	2	6		2	4	3	
Total	**23**	**17**	**19**	**35**	**17**	**26**	

Theme	Activity Summary
• Manual Detection • - 2 out of 4 manual detection in scope for CIAO (Orchestration)	CIAO alerts – Need timeline for enabling as Sev -2
• Manual Detection • - 2 out of 4 manual detection in scope for CIAO (Orchestration)	CIAO alerts – Need timeline for enabling as Sev -2
• Several Datacenters running hot (not a direct cause of incidents) -Some capacity coming online in November - Ongoing risk to mitigation of future incidents	• Migrating services out of SCUS where possible. • Additional capacity coming online in November in SCUS
• Capacity - 9 total incidents due to capacity - 5 repeat capacity incidents (SignalR and AutoScale)	• Fix added to more aggressively auto scale • Customer impact overstated • SignalR commands whitelisted • 3 week window from initial incident to SignalR









Application Incidents - 16

Count	Category	Comments
9	Capacity	5 - Auto Scale and SignalR 3 - Unexpected load 1 - SQL Azure right sizing
5	Code Defect	Code introduced from deployment/config change
1	Data Shape	SQL query plan regression
1	Other	Planned MSENG re-parenting

Kaizen Review

Incident response by month
- Availability (per customer)
- Automated detection
- Time to detect
- Time to mitigate

Every incident that affected users
- Identify patterns
- Ensure remediation

All customer support inquiries
- Reduce incoming/000 users

Costs of operation
- Reduce $/000 users

Habits we've learned so far at Microsoft

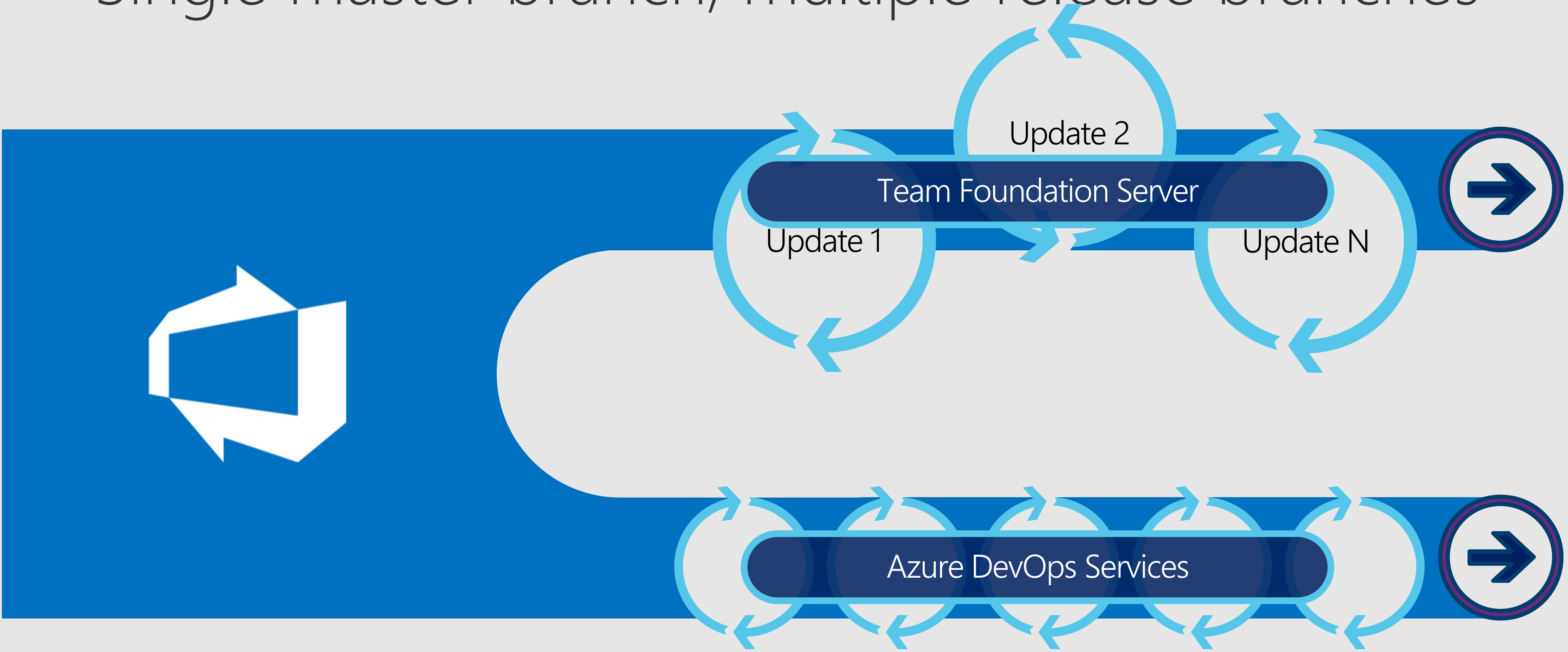
- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- **Infrastructure as Flexible Resource**
- Don't over-think, learn how to fail fast

Code: Cloud first, then move on-premises

One code base with multiple delivery streams

Shared abstraction layer

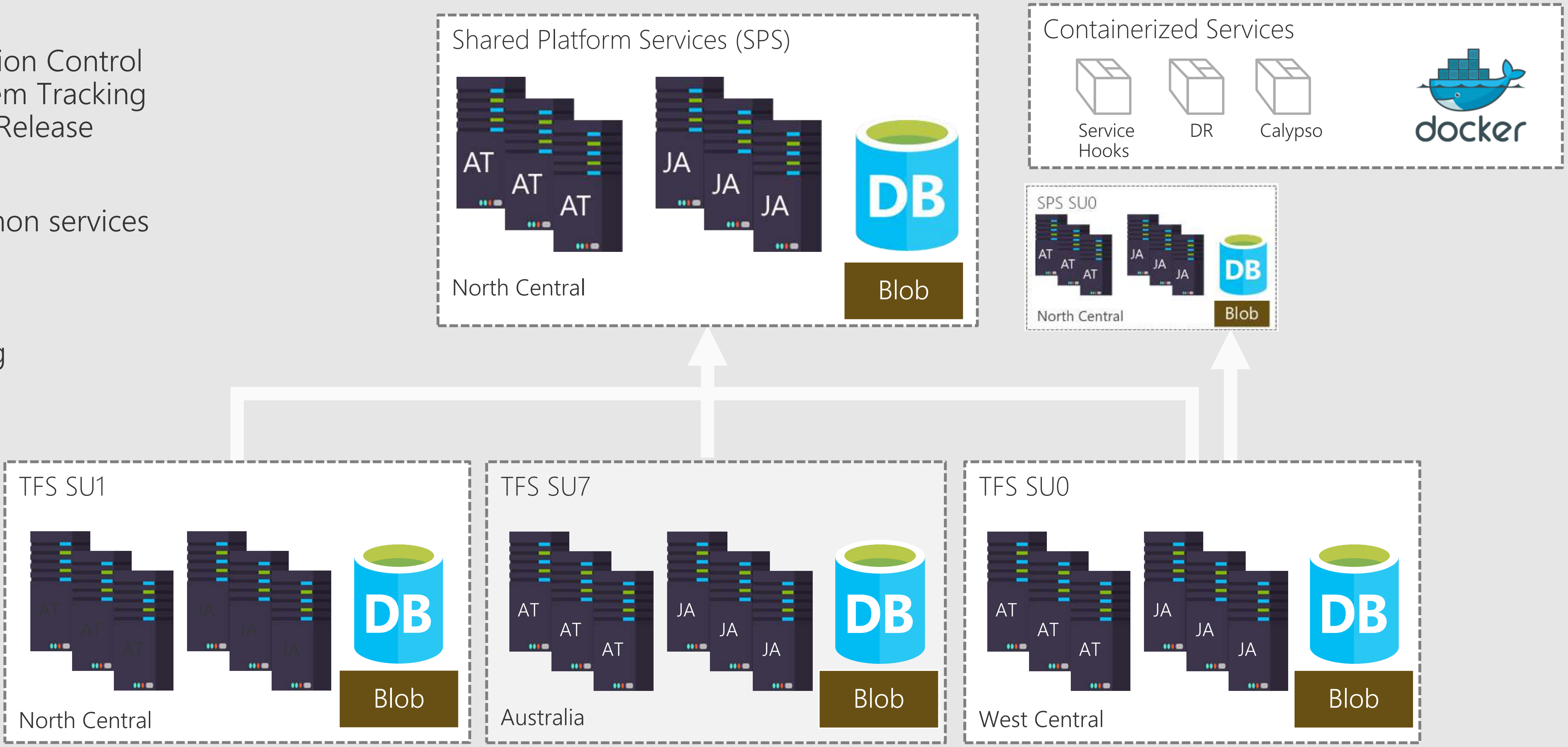
Single master branch, multiple release branches



Multiple Data Centers with incremental roll out

TFS
- Git/Version Control
- Work Item Tracking
- Build & Release
- Test

SPS: common services
- Account
- Identity
- Profile
- Licensing



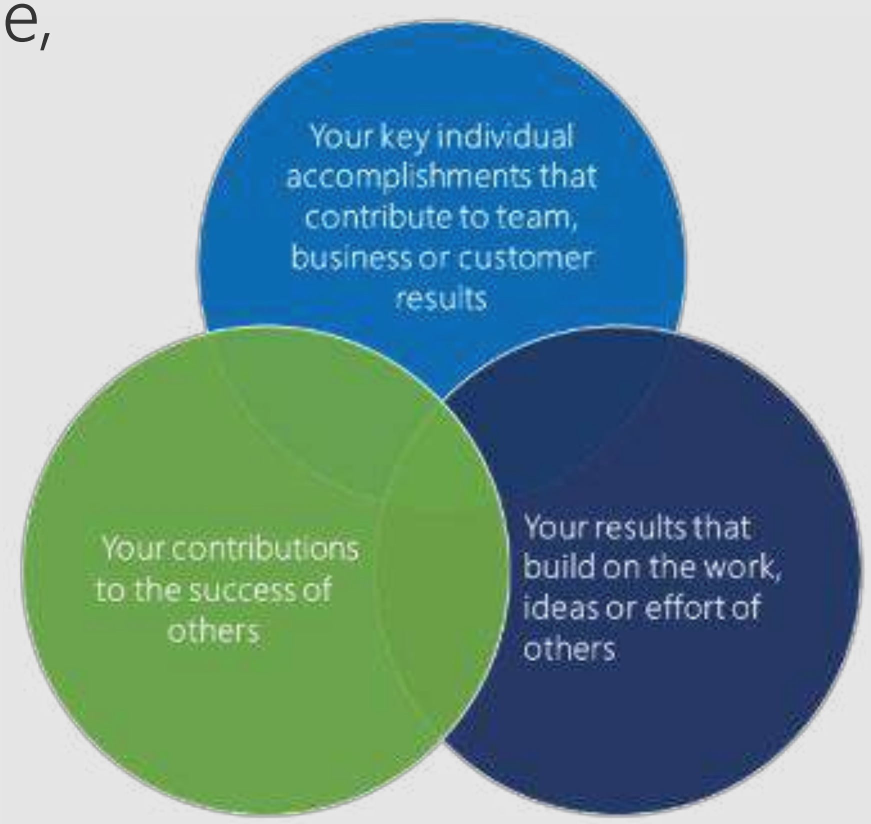
Habits we've learned so far at Microsoft

- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

Sharing, not siloes

High-performing teams thrive when the culture enables inner sourcing - the sharing of knowledge, skills and code inside the organization.

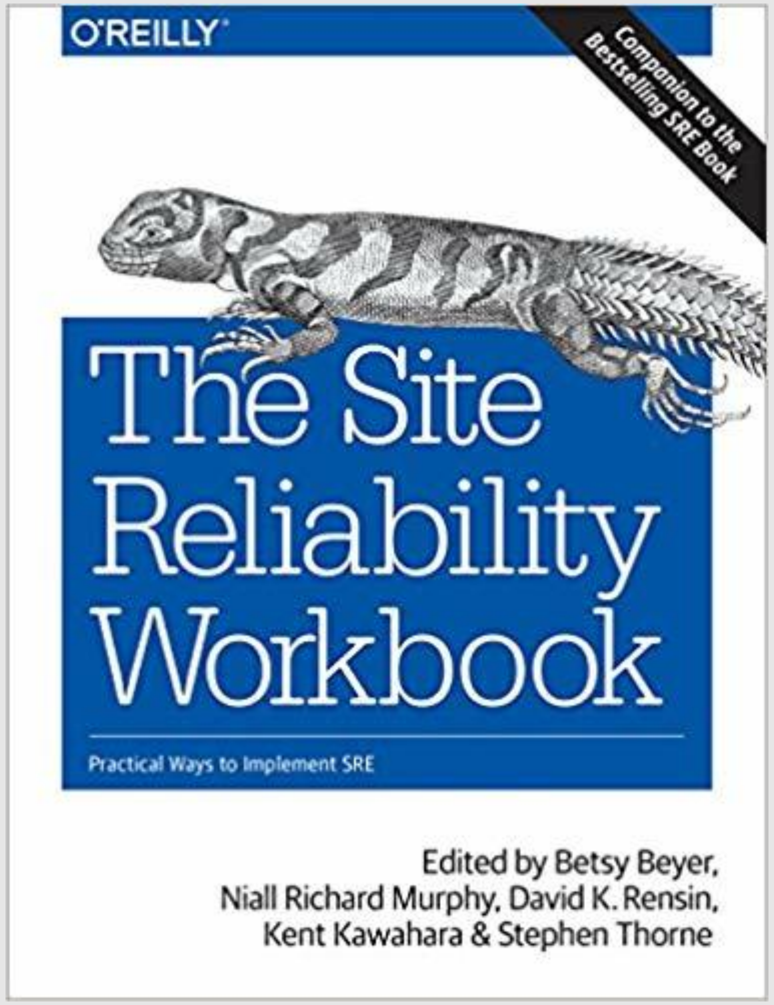
- Changed incentives to encourage sharing
- Made sharing with the org the default
- Support cross-org fork and pull request workflows


Your key individual accomplishments that contribute to team, business or customer results
Your contributions to the success of others
Your results that build on the work, ideas or effort of others

Measuring Impact at Microsoft

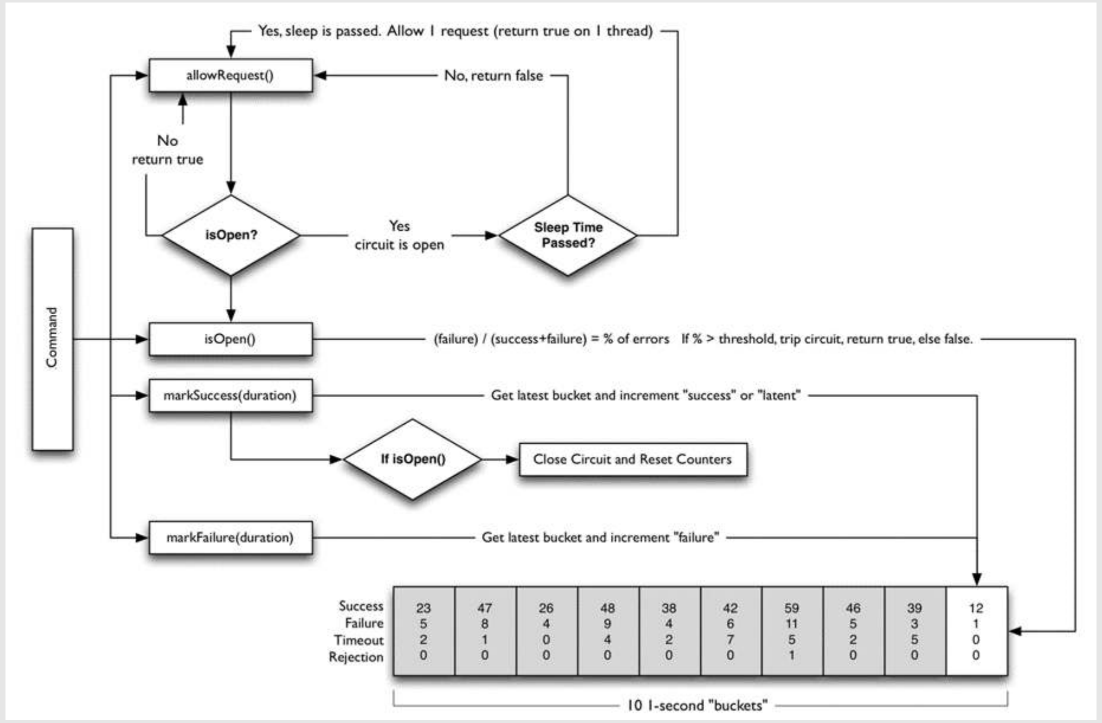
Building the SRE Discipline

- Full career track for Site Reliability Engineers with grades all the way up to CVP equivalent in the business.
- Investing in blameless but details and actionable post- mortems that span the business (DC / network / app)
- Incorporate human factors into safety systems


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

Resilience Engineering

- Design with failure in mind
- Circuit breakers
- Self healing systems
- Safely introducing faults to test resilience



Yes, sleep is passed. Allow 1 request (return true on 1 thread)
allowRequest()
No, return false
No
return true
isOpen?
Yes
circuit is open
Sleep Time Passed?
Command
isOpen()
(failure) / (success+failure) = % of errors If % > threshold, trip circuit, return true, else false.
markSuccess(duration)
Get latest bucket and increment "success" or "latent"
If isOpen()
Close Circuit and Reset Counters
markFailure(duration)
Get latest bucket and increment "failure"
Success 23 47 26 48 38 42 59 46 39 12
Failure 5 8 4 9 4 6 11 5 3 1
Timeout 2 1 0 4 2 7 5 2 5 0
Rejection 0 0 0 0 0 0 1 0 0 0
10 1-second "buckets"

- Goal: Chaos engineering by default across Azure services

Microsoft DevOps Transformation

The story so far…

➔ https://aka.ms/DevOpsAtMicrosoft

Before	After
4-6-month milestones	3-week sprints
Horizontal teams	Vertical teams
Personal offices	Team rooms
Long planning cycles	Continual Planning and Learning
PM, Dev, Test	PM and Engineering
Yearly customer engagement	Continual customer engagement
Feature branches	Everyone in master
20+ person teams	8-12 person teams
Secret roadmap	Publicly shared roadmap
Bug debt accumulated	Debt paid as incurred
100-page spec documents	Mockups in PPT
Private repositories	Inner source
Deep organizational hierarchy	Flattened organization hierarchy
Success is a measure of install numbers	User satisfaction determines success
Features shipped once a year	Features shipped every sprint

A journey of a thousand miles begins with a single sprint

Microsoft Azure

DevOps Dojo

DevOps Dojo Approach

SHU

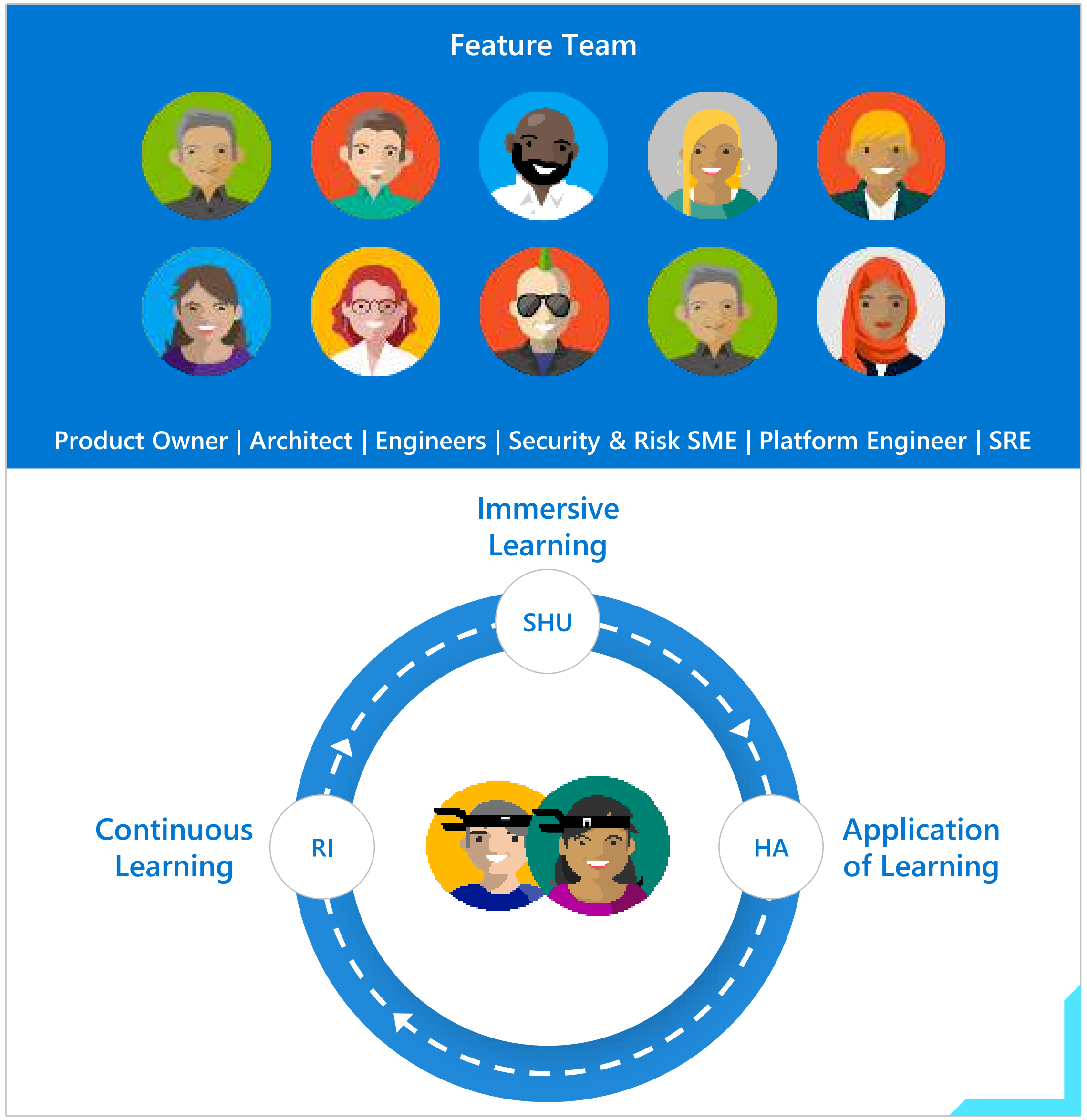
In this beginning stage the student follows the teachings of one master precisely. He/she concentrates on how to do the task, without worrying too much about the underlying theory. If there are multiple variations on how to do the task, he/she concentrates on just the one way his master teaches him.

HA

At this point the student begins to branch out. With the basic practices working he/she now starts to learn the underlying principles and theory behind the technique. He also starts learning from other masters and integrates that learning into his practice.

RI

Now the student isn't learning from other people, but from his own practice. He creates his own approaches and adapts what he's learned to his own circumstances.


Feature Team
Product Owner | Architect | Engineers | Security & Risk SME | Platform Engineer | SRE
Immersive Learning
SHU
Continuous Learning
RI
HA
Application of Learning

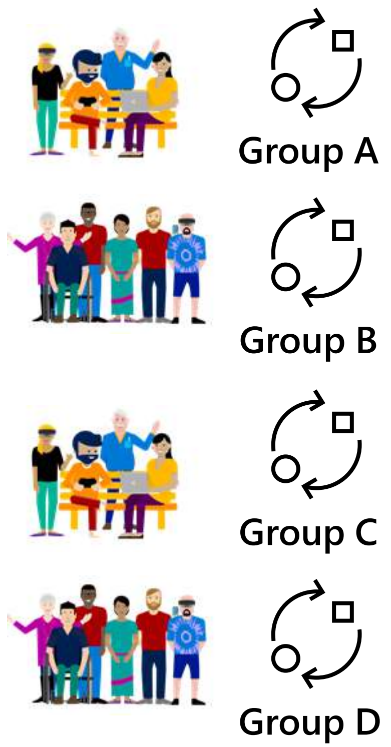
Immersive Dojo Master Class

"Tell me and I forget. Teach me and I remember. Involve me and I learn." – Benjamin Franklin

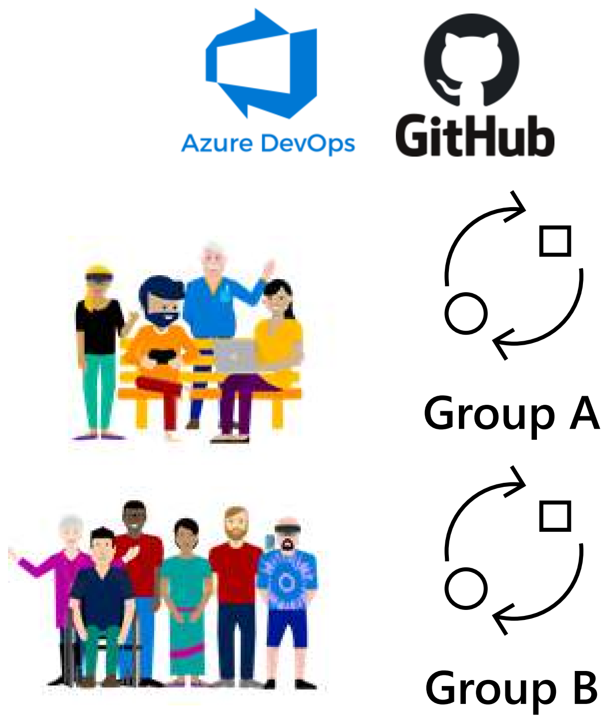
1 Hour: Theory/Concept

Point of View Deck
(1 Team)

1 Hour: Discussion/Design

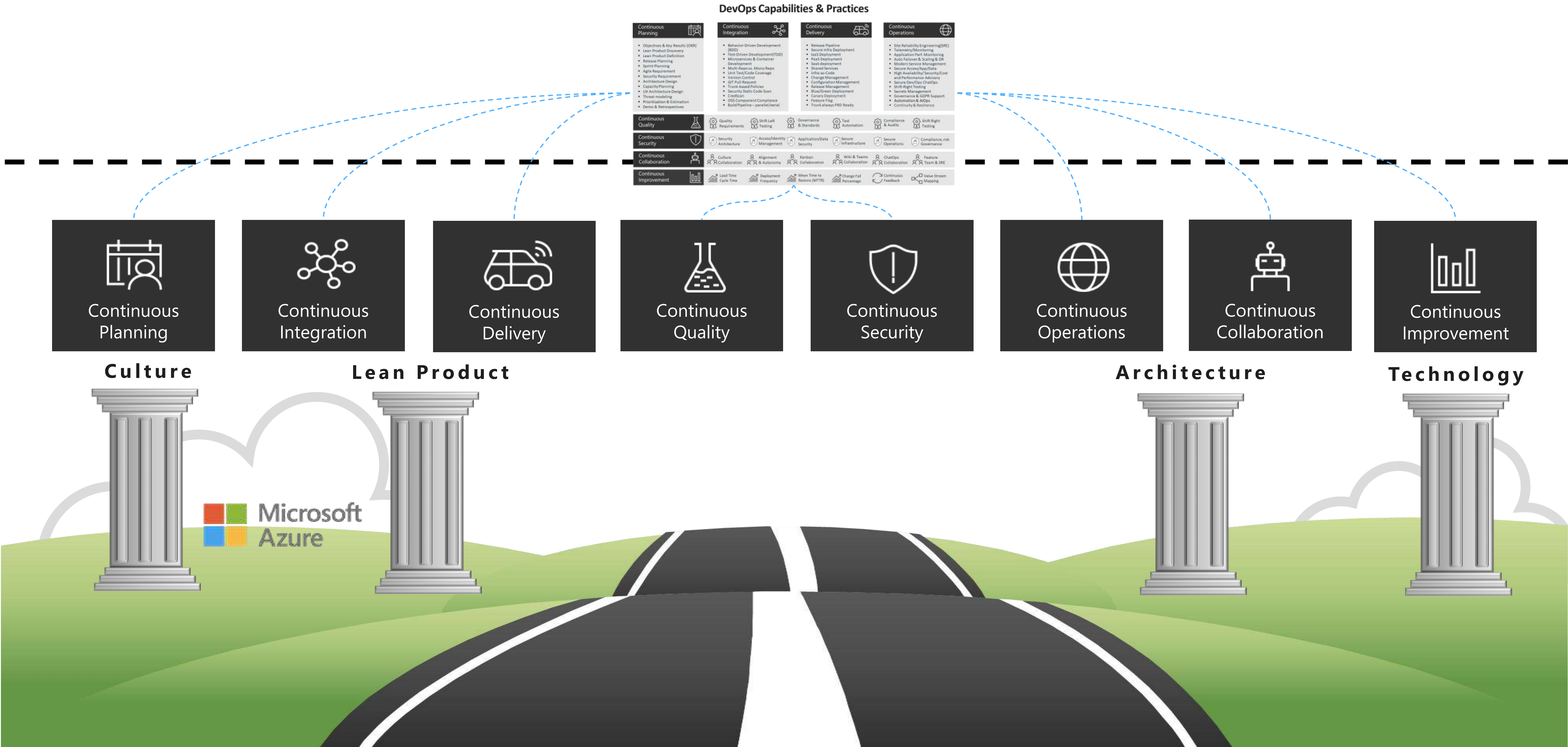


1 Hour: Hands-on/Labs



Continuous Planning	Continuous Integration	Continuous Delivery	Continuous Quality	Continuous Security	Continuous Operations	Continuous Collaboration	Continuous Improvement

Dojo White Belt



DevOps Capabilities & Practices

Continuous Planning

- Objectives & Key Results (OKR)
- Lean Product Discovery
- Lean Product Definition
- Release Planning
- Sprint Planning
- Agile Requirement
- Security Requirement
- Architecture Design
- Capacity Planning
- UX Architecture Design
- Threat modeling
- Prioritization & Estimation
- Demo & Retrospectives

Continuous Integration

- Behavior-Driven Development (BDD)
- Test-Driven Development(TDD)
- Microservices & Container Development
- Multi-Repo vs. Mono Repo
- Unit Test/Code Coverage
- Version Control
- GIT Pull Request
- Trunk-based Policies
- Security Static Code Scan
- CredScan
- OSS Component Compliance
- Build Pipeline – parallel/serial

Continuous Delivery

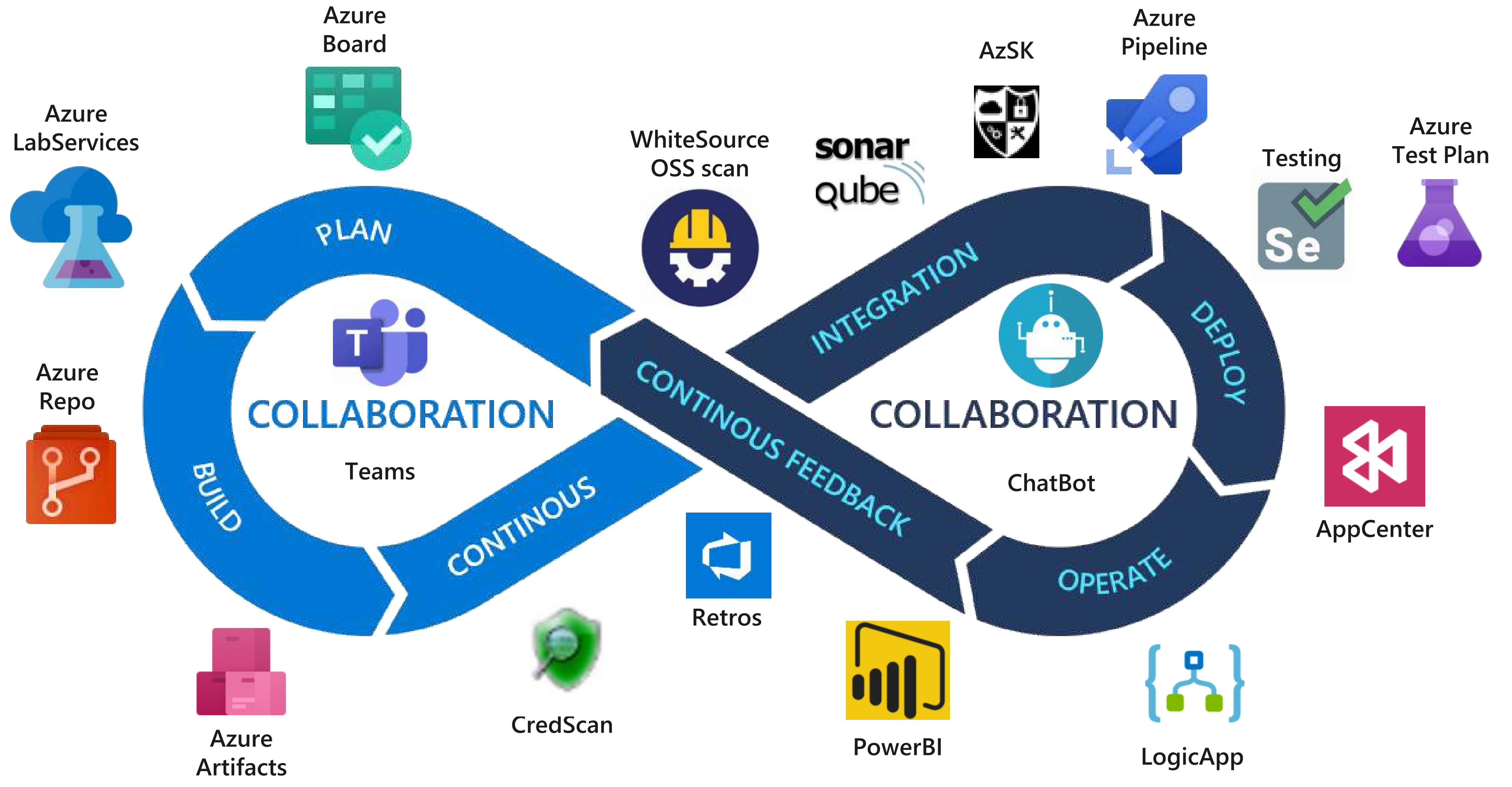
- Release Pipeline
- Secure Infra Deployment
- IaaS Deployment
- PaaS Deployment
- SaaS deployment
- Shared Services
- Infra-as-Code
- Change Management
- Configuration Management
- Release Management
- Blue/Green Deployment
- Canary Deployment
- Feature Flag
- Trunk always PRD Ready

Continuous Operations

- Site Reliability Engineering(SRE)
- Telemetry/Monitoring
- Application Perf. Monitoring
- Auto Failover & Scaling & DR
- Modern Service Management
- Secure Access/App/Data
- High Availability/ Security/Cost and Performance Advisory
- Secure Dev/Ops ChatOps
- Shift-Right Testing
- Secrets Management
- Governance & GDPR Support
- Automation & AIOps
- Continuity & Resilience

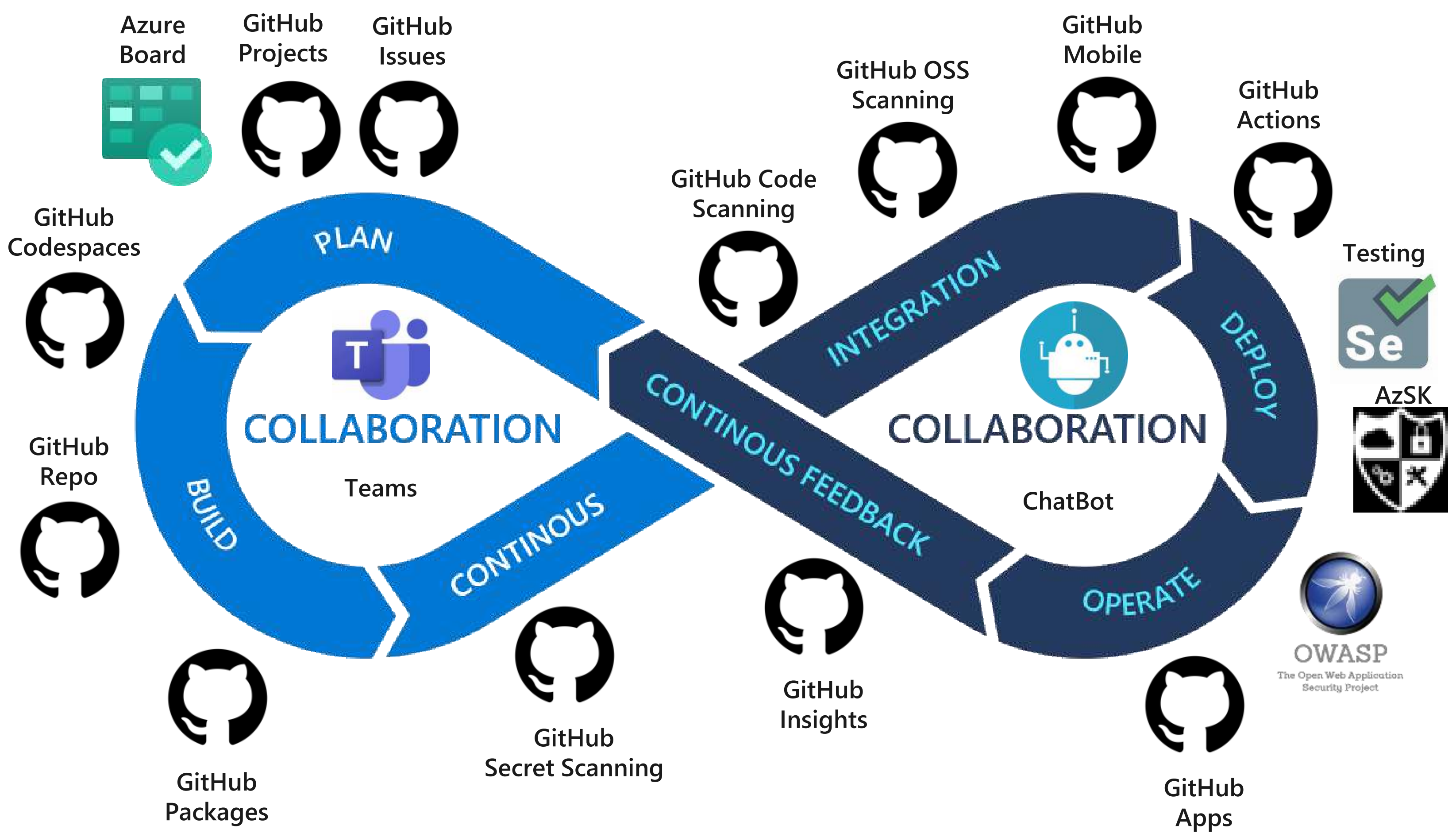
Continuous Quality	Quality Requirements	Shift Left Testing	Governance & Standards	Test Automation	Compliance & Audits	Shift Right Testing
Continuous Security	Security Architecture	Access/Identity Management	Application/Data Security	Secure Infrastructure	Secure Operations	Compliance, risk Governance
Continuous Collaboration	Culture Collaboration	Alignment & Autonomy	Kanban Collaboration	Wiki & Teams Collaboration	ChatOps Collaboration	Feature Team & SRE
Continuous Improvement	Lead Time Cycle Time	Deployment Frequency	Mean Time to Restore (MTTR)	Change Fail Percentage	Continuous Feedback	Value Stream Mapping

Azure DevOps – Labs reference implementation



Microsoft Azure

GitHub- Labs reference implementation



Microsoft Azure

More Information about

DevOps Dojo

[Intro of DevOps Dojo - Azure DevOps Blog (microsoft.com)]

Do you want to hear more around Agile / DevOps?

Register here for our next Azure Operation Roundtable
" How to operate a Digital Product "
15.11.2022 – 14:00 – 16:30
@Circle

" How to operate a Digital Product "

Microsoft

Thank you!

http://aka.ms/MSDevOps

Kyle Krüsi
kyle.kruesi@microsoft.com