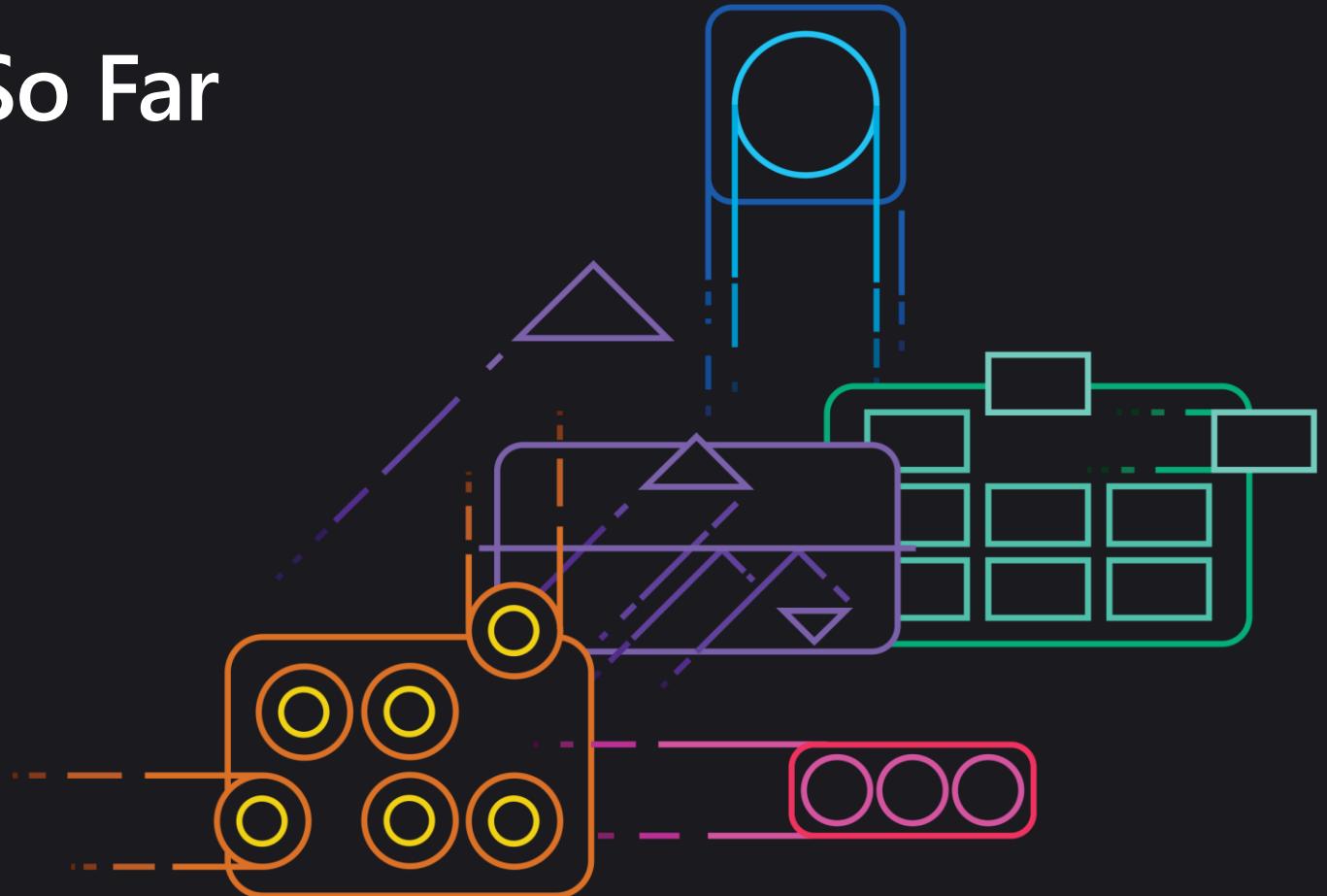


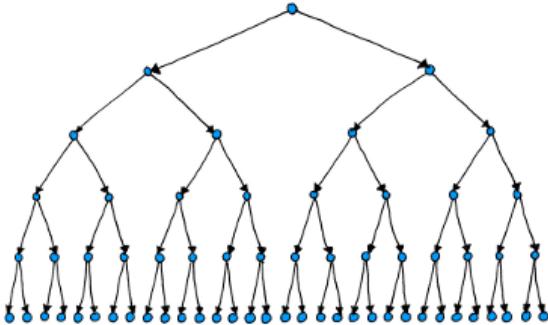
The DevOps Journey So Far at Microsoft

Michael Royster

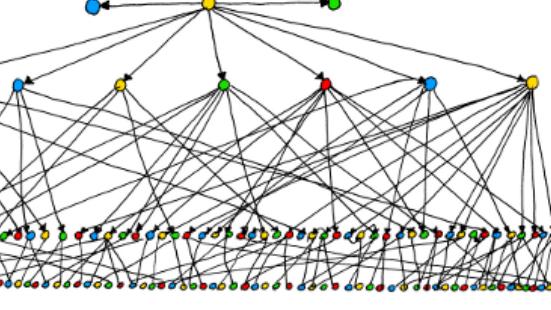
Cloud Solution Architect
Microsoft Ltd
mroyster@microsoft.com



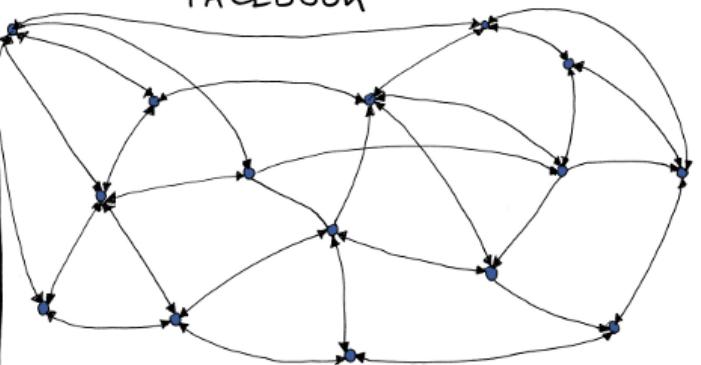
AMAZON



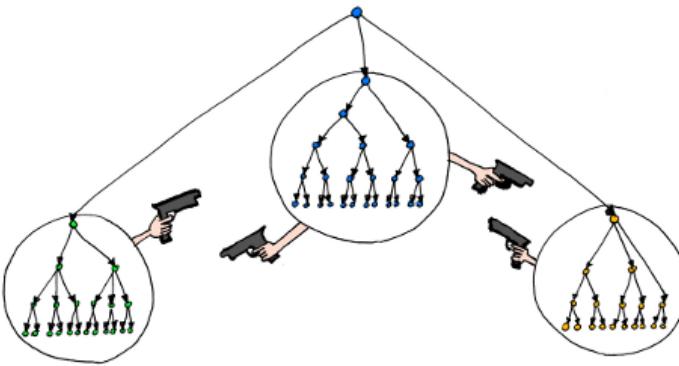
GOOGLE



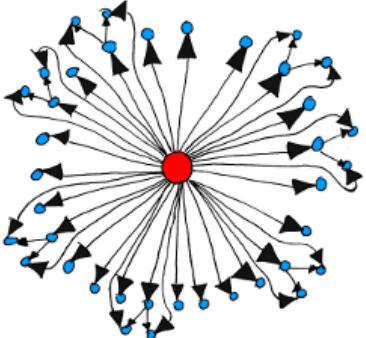
FACEBOOK



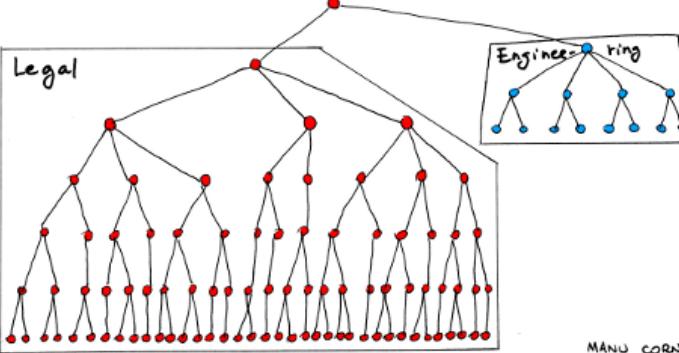
MICROSOFT



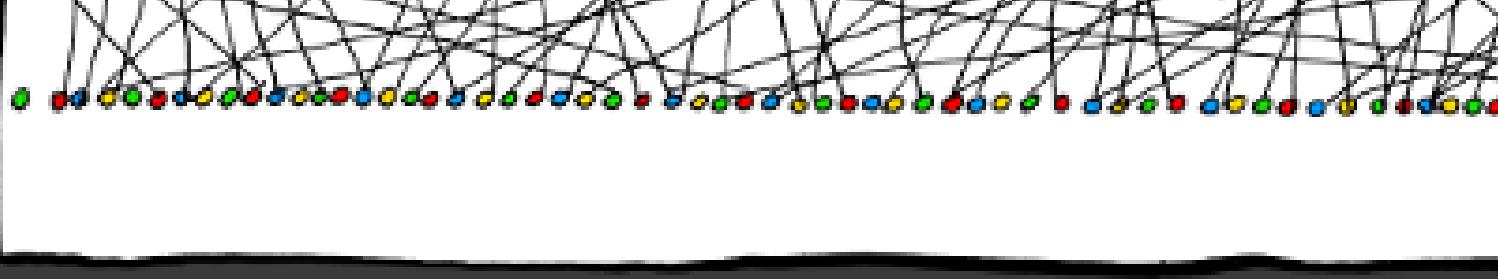
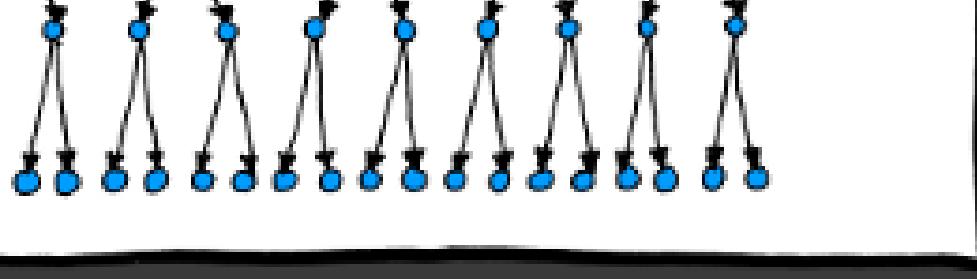
APPLE



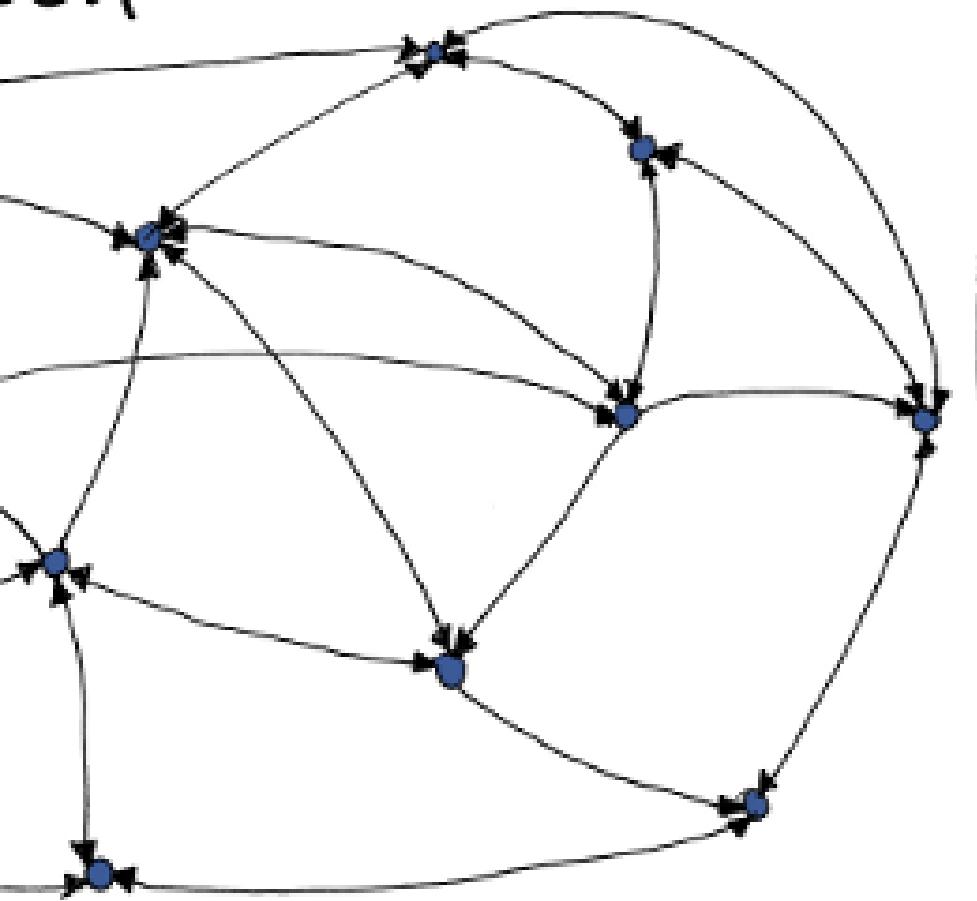
ORACLE



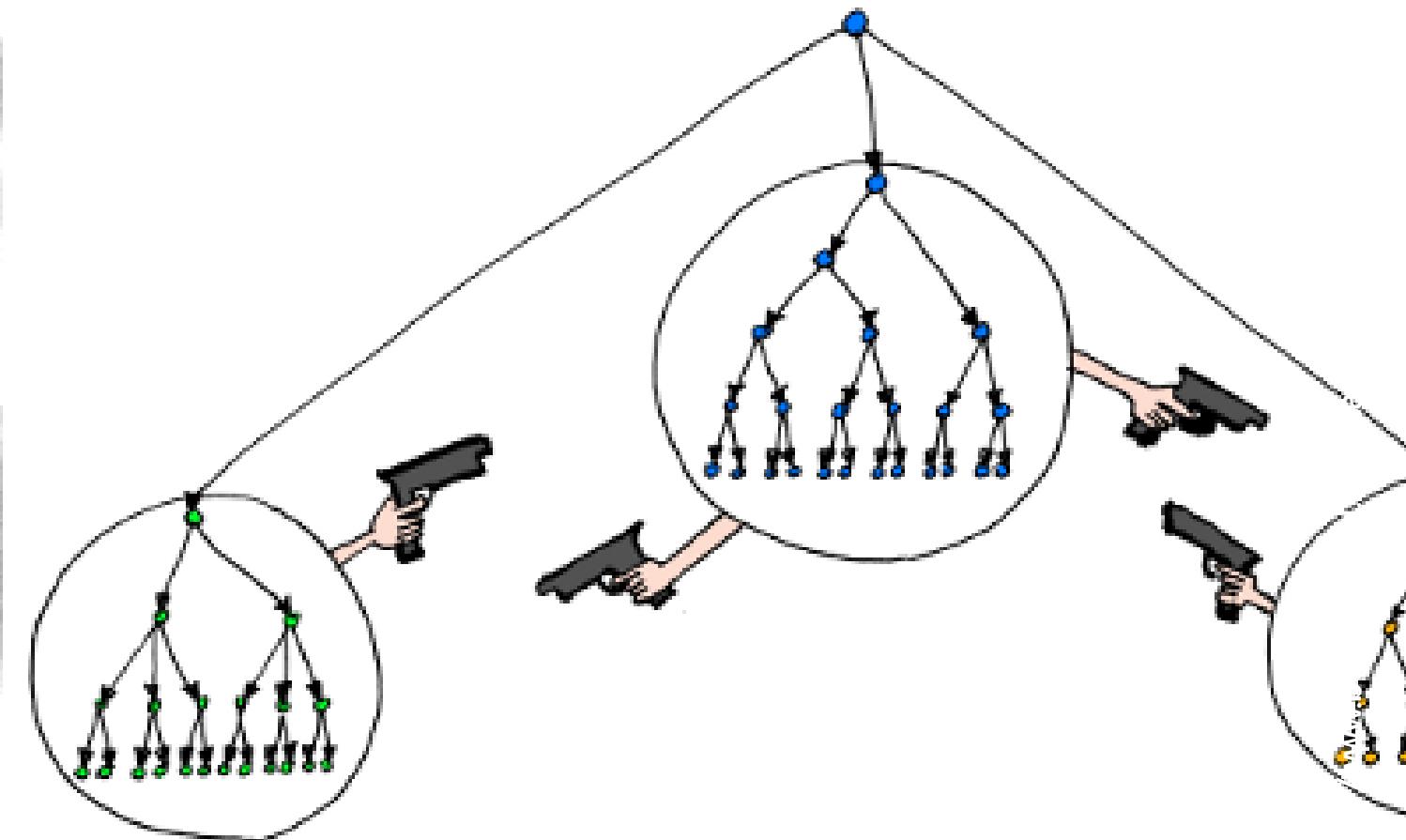
MANU CORNET



book



MICROSOFT





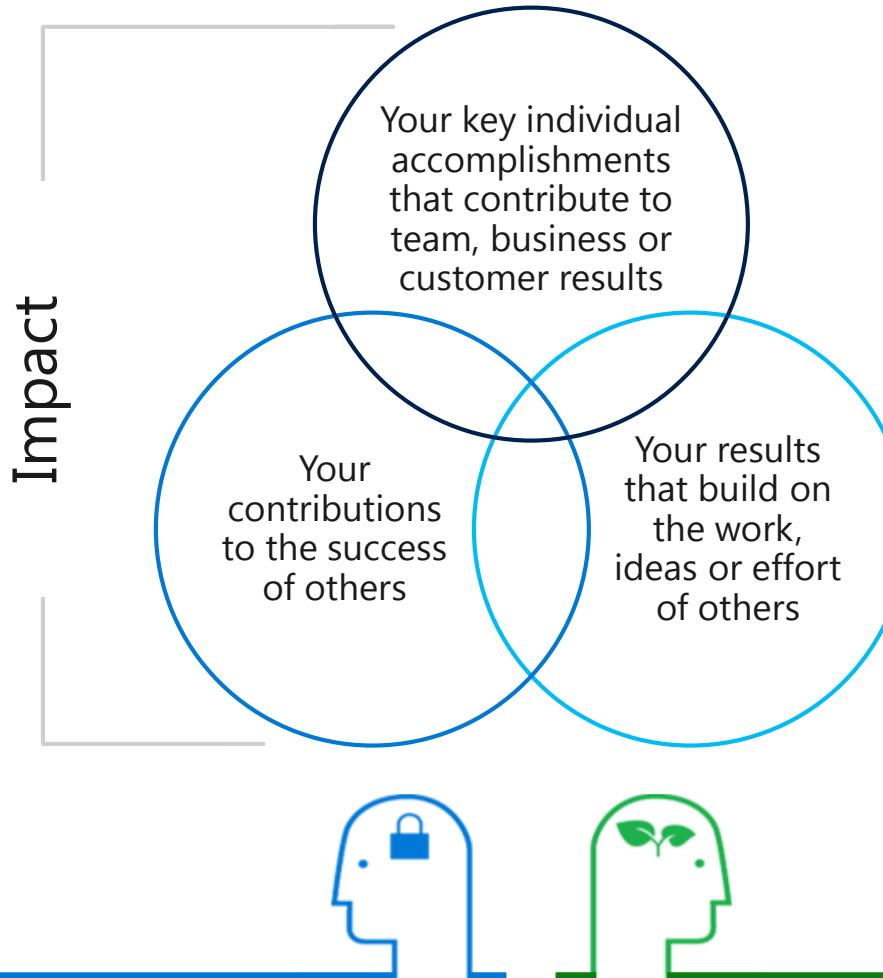
Growth mindset



Customer obsessed
Diverse & Inclusive → Make a difference
One Microsoft

Rewards: from activity to impact

In the spirit of a growth mindset, while we perform and deliver impact, we also want to transform—by learning, finding lessons and inspiration in the success of others, self-reflecting and really hearing feedback from others, embracing challenges and risks and using our ‘misses’ to grow more masterful—this is how we continually improve.



Fixed Mindset leads to desire to look smart and therefore tendency to avoid challenges, give up easily, see failures as fruitless or worse, ignore useful negative feedback and feel threatened by success of others.



Growth Mindset leads to a desire to learn and therefore a tendency to embrace challenges, persist in the face of setback, see failure as essential to mastery, learn from criticism and find lessons and inspiration in the success of others.

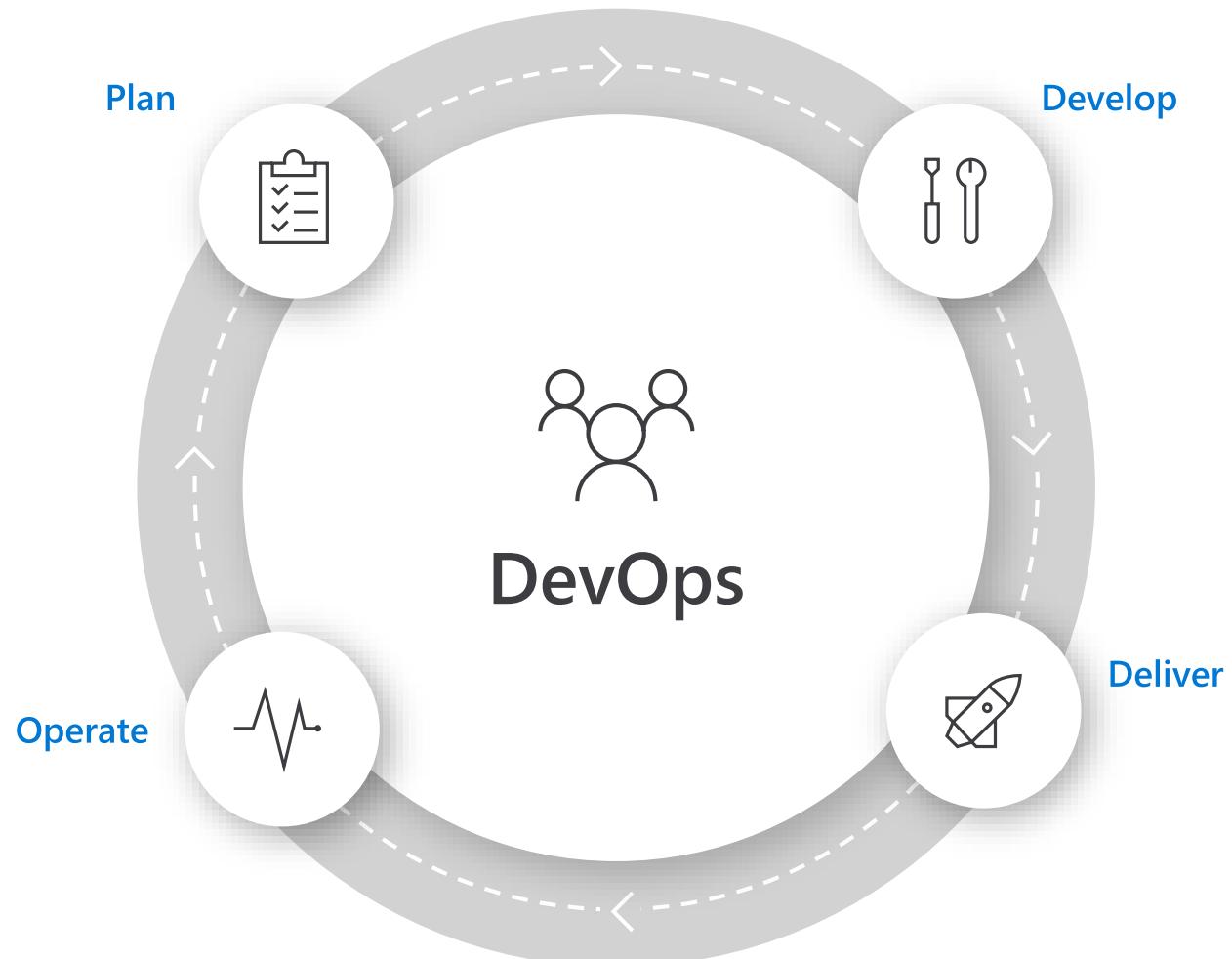


Modern app engineering is enabled by DevOps

“

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

”

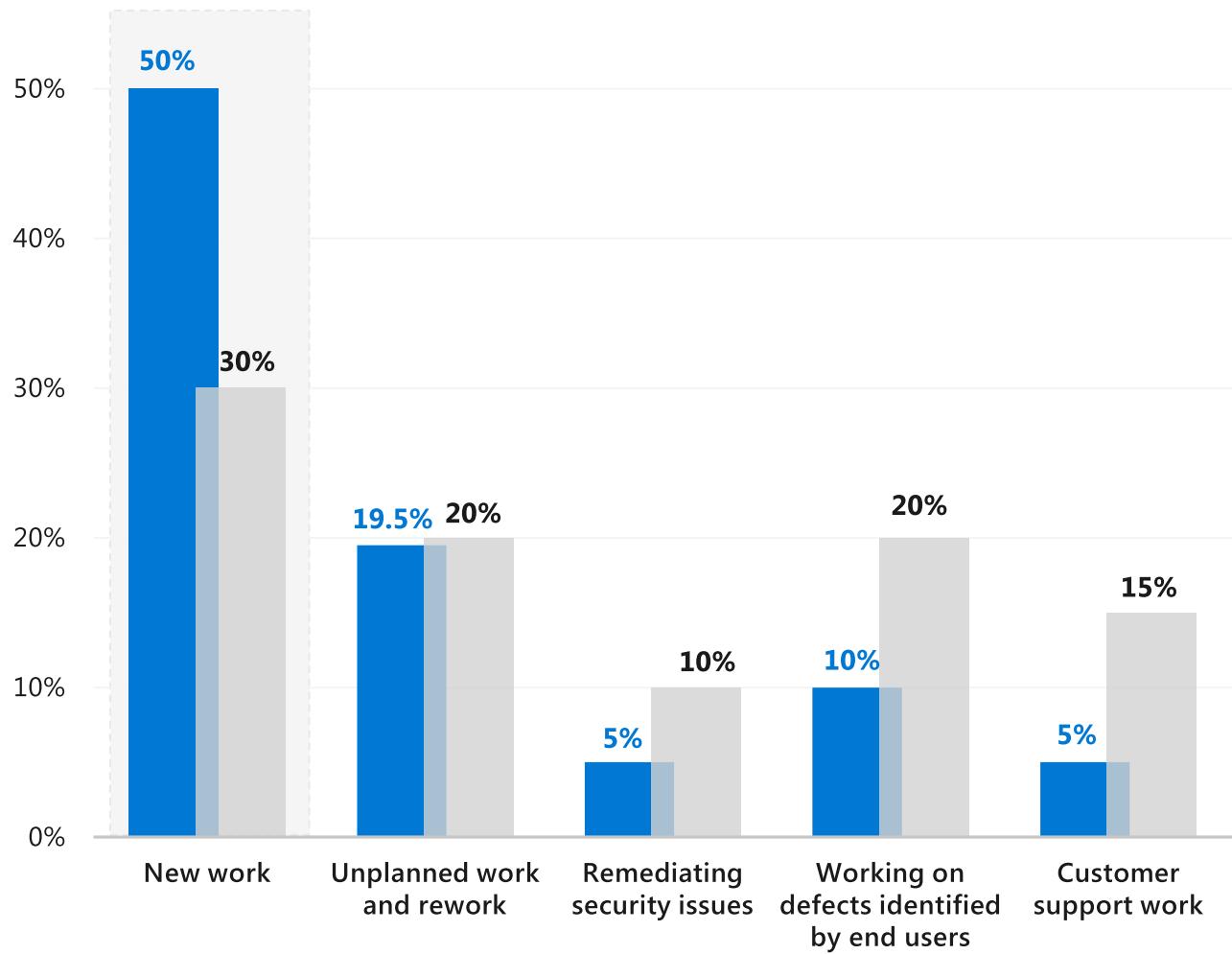


Innovation with oversight

Top performing DevOps companies spend more time innovating and less time “keeping the lights on”.

The result: better products, delivered faster, to happier customers by more engaged teams

Time spent



Effective Enterprise DevOps

Five key practices

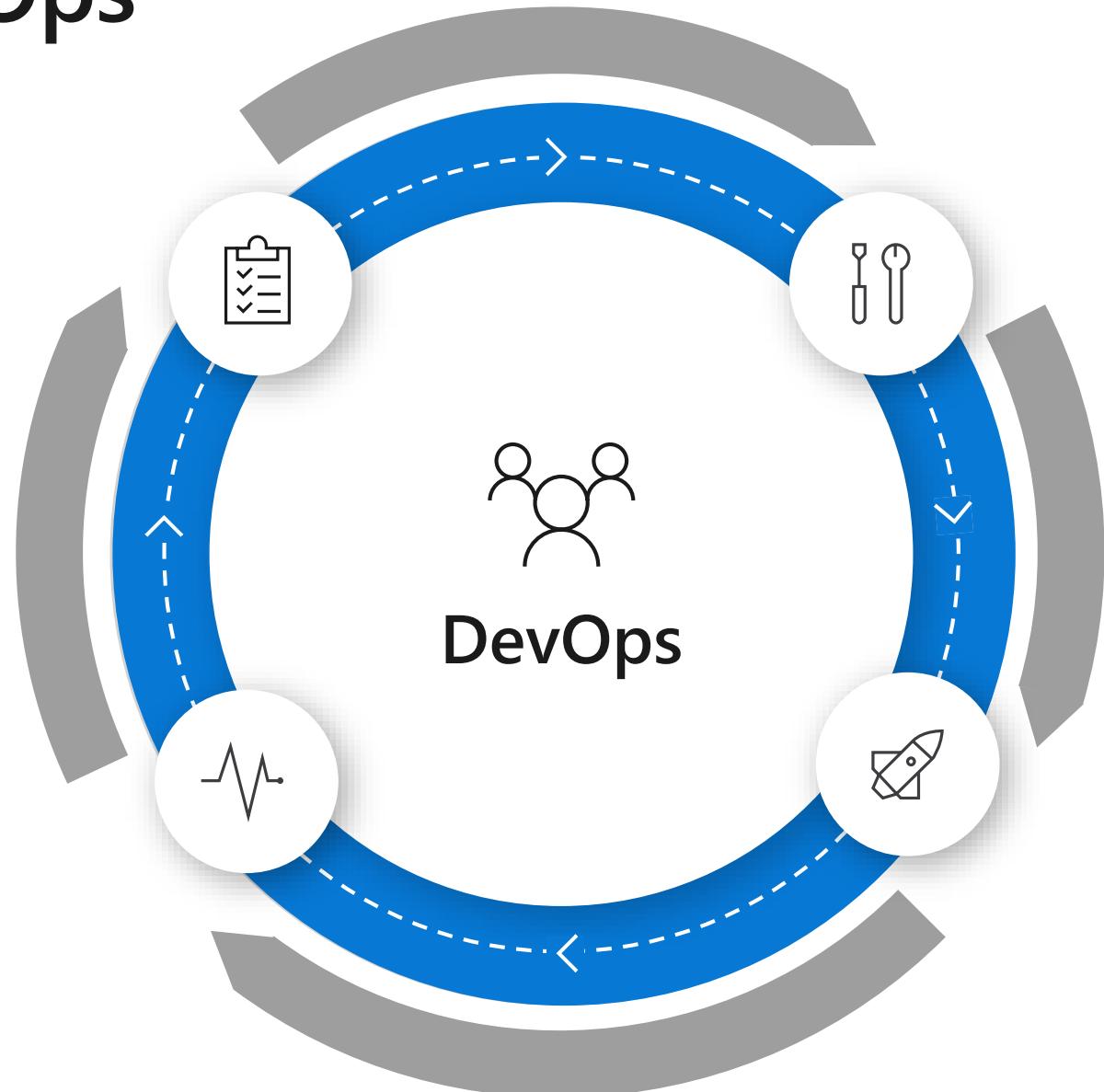
Continuous Improvement

Collaboration

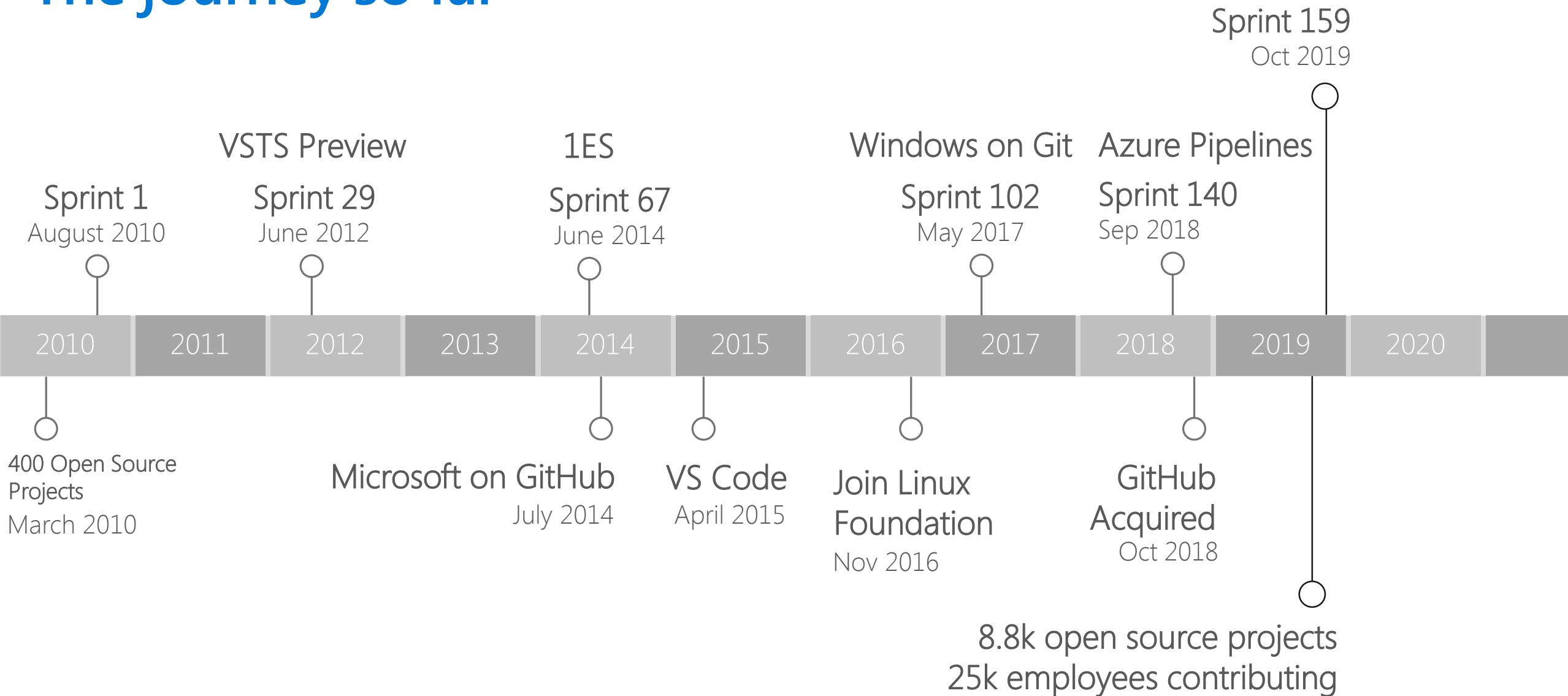
Workflow Automation

Security & Compliance

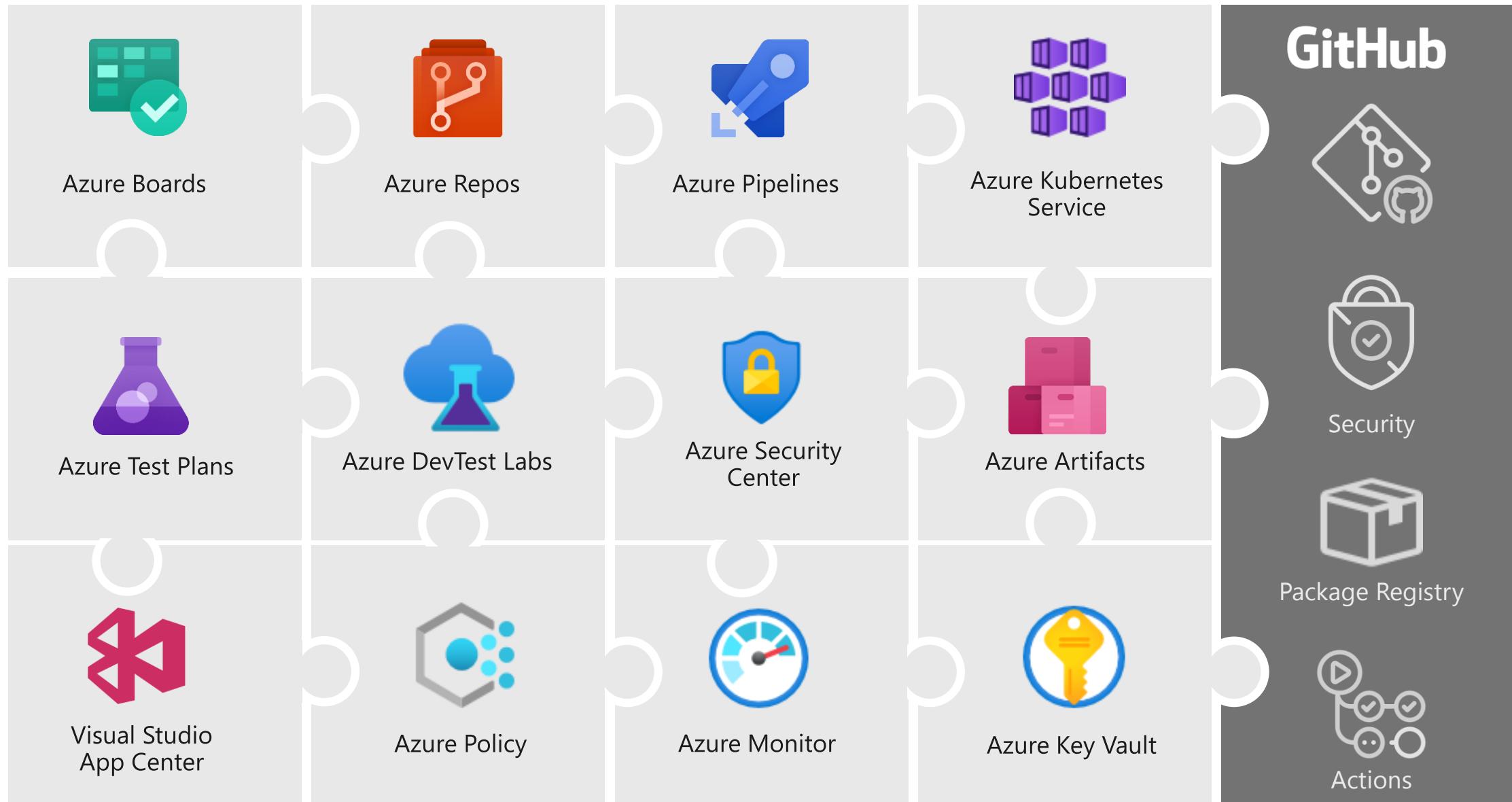
Monitoring & Learning



The journey so far



Microsoft's DevOps Tooling – enhanced by GitHub



DevOps at Microsoft

→ <https://aka.ms/DevOpsAtMicrosoft>

110k

Active users inside
Microsoft

4.6m

Builds per month

28k

Work items
created per day

2.4m

Private Git commits per
month

8.8k

Open Source repos on
GitHub

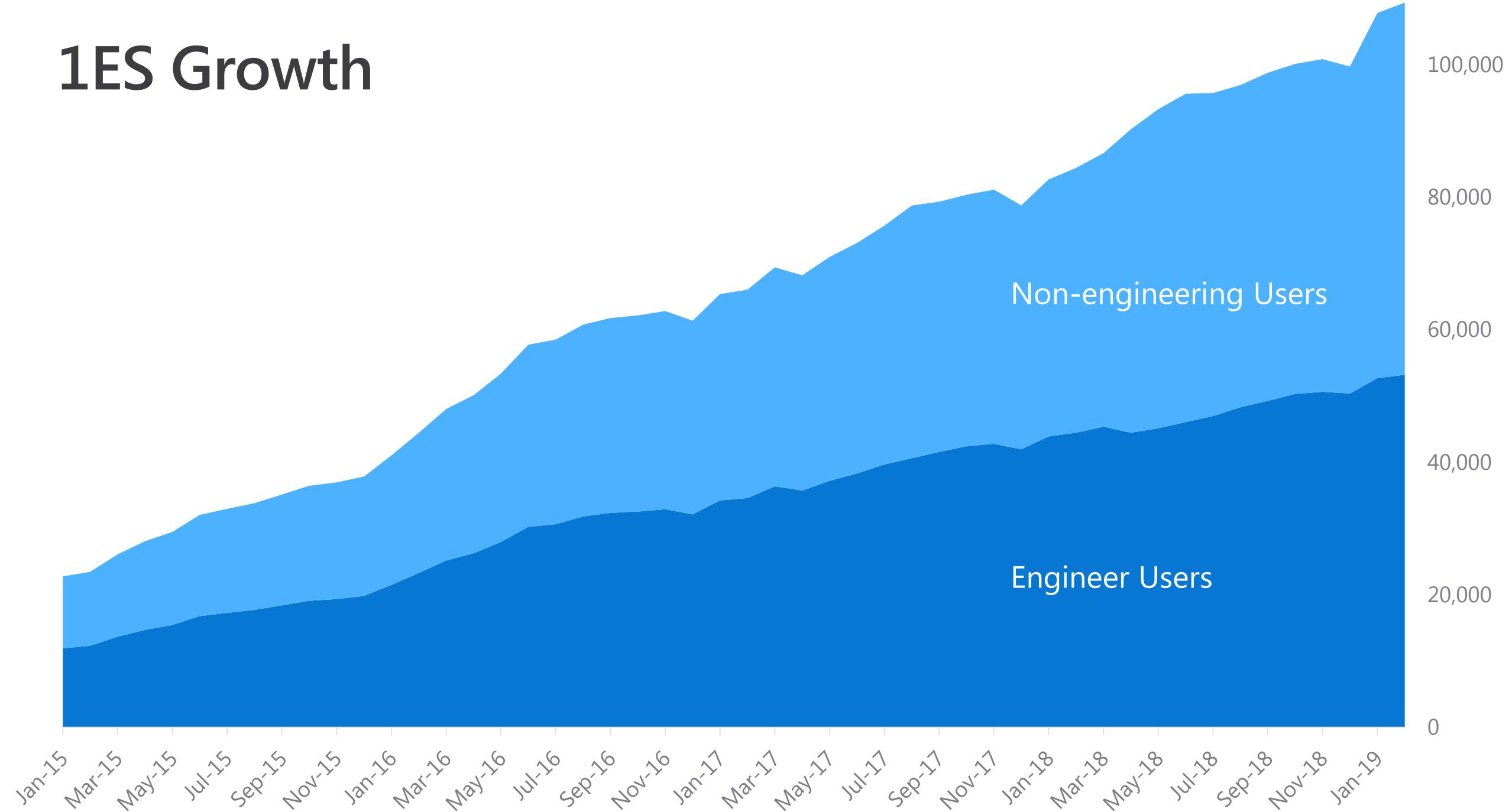
25k

Employees contributing
to open source

82,000

Deployments per day

1ES Growth





107,000

Engineers in Microsoft
working with Azure
DevOps

One second per day is like
adding

3.7 More people to
Microsoft

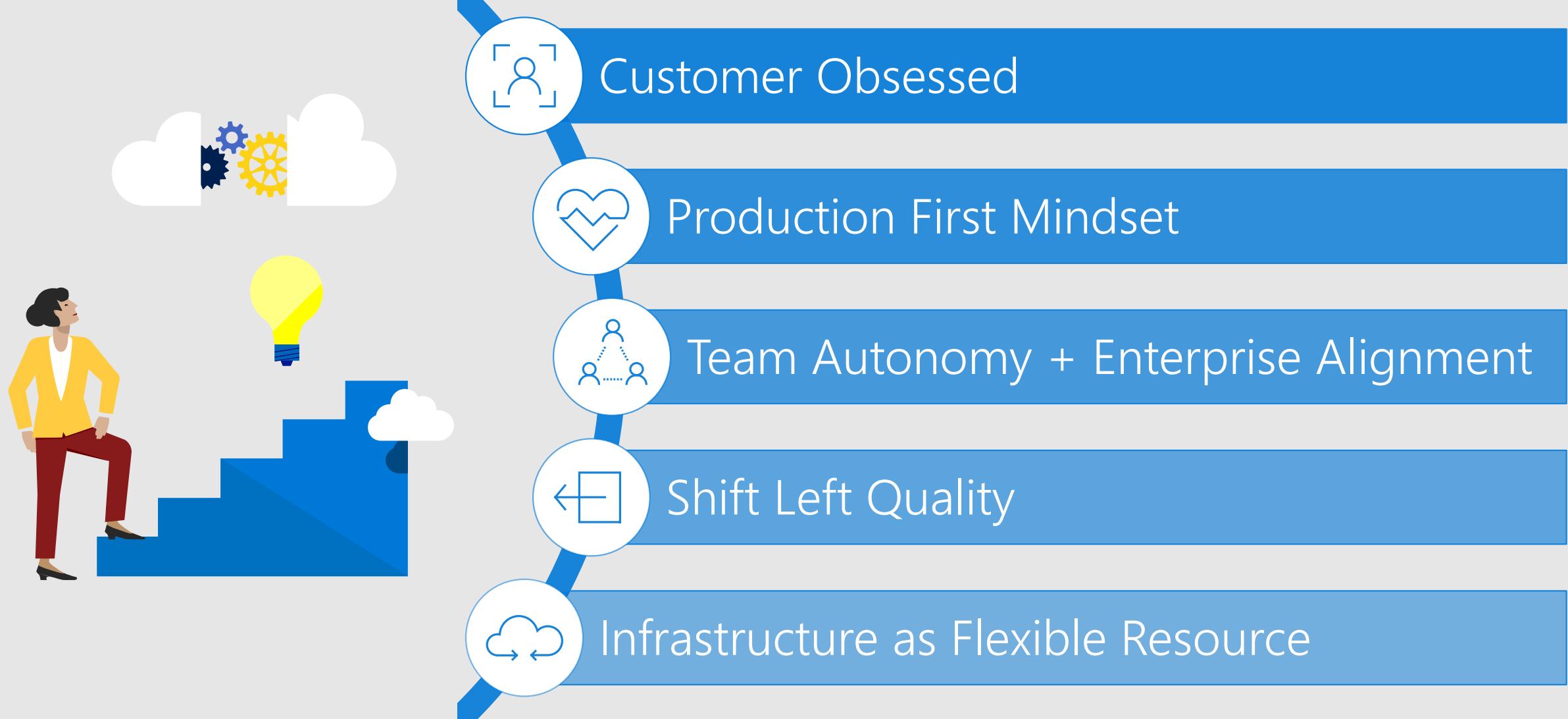
One minute per day is like
adding

163 More people to
Microsoft

One hour per day is like
adding

\$2.7B
per year

Five habits we've learned so far



Five habits we've learned so far



Listen to our customers

Quantitatively & Qualitatively

Developer Community
Get help from our community supported forum

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

Stack Overflow [azure-devops]

Questions tagged [azure-devops]

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

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

Hot Network Questions

Longest word chain from a list of words

Word for a person responsible for collecting and

Info Newest Featured Frequent Votes Active Unanswered

7,330 questions

3 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 Quiver 610 2 12 32 6k views modified 1 hour ago

1 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 Martin Brandl 34.2k 10 48 84 454 views modified 1 hour ago

1 Build sqiproj 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

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

A AzureDevOps

About this project

Azure DevOps helps you plan smarter, collaborate better, and ship faster with a modern set of developer 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)

We want your feedback!

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

1 2 3 4 5 6 7 8 9 10

NOT AT ALL LIKELY EXCELLENTLY LIKELY

Location Business Champ Engaged Users ▲ MoM Boards ▲ Repos ▲ Azure C

Location	Business	Champ	Engaged Users	▲ 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%	56.5%	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%

Our Definition of Done

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



Build-Measure-Learn

Hypothesis



We believe {customer segment} wants {product/feature} because {value prop}

Experiment



To prove or disprove the above, the team will conduct the following experiment(s): ...

Learning



The above experiment(s) proved or disproved the hypothesis by impacting the following metric(s): ...

But measure what's important (KPI's)

Usage

- Acquisition
- Engagement
- Satisfaction
- Churn
- Feature Usage

Velocity

- Time to Build
- Time to Self Test
- Time to Deploy
- Time to Learn

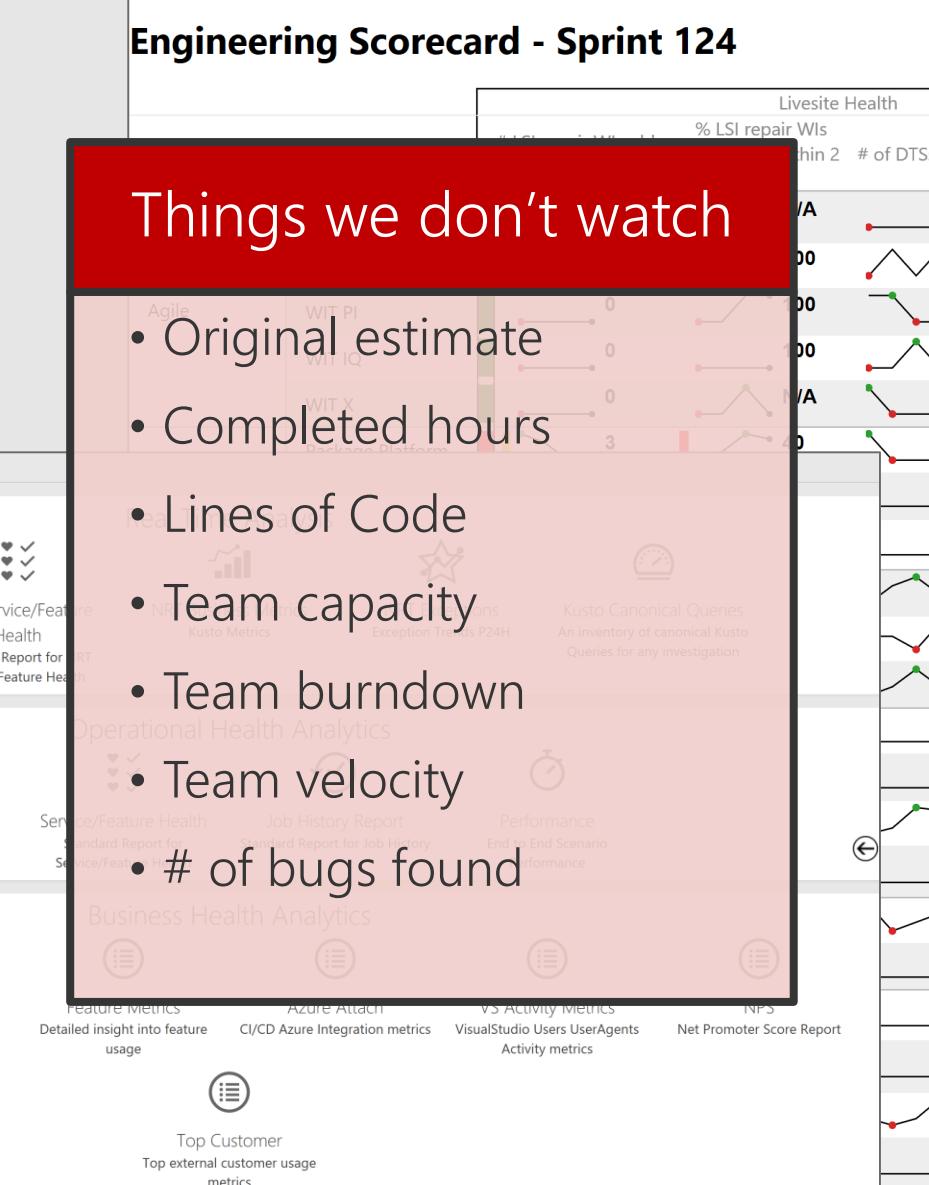
Live Site Health

- Time to Detect
- Time to Communicate
- Time to Mitigate
- Customer Impact
- Incident Prevention Items
- Aging Live Site Problems
- SLA per Customer
- Customer Support Metrics

Engineering Scorecard - Sprint 124

Things we don't watch

- Original estimate
- Completed hours
- Lines of Code
- Team capacity
- Team burndown
- Team velocity
- # of bugs found



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)

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)

Data Size

PAST WEEK • REFRESHED: 8:08:01 AM

47,686.9

Avg Total Data Size (TB)

Compression Ratio

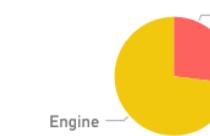
PAST WEEK • REFRESHED: 8:08:01 AM

7.51

Comp. Ratio

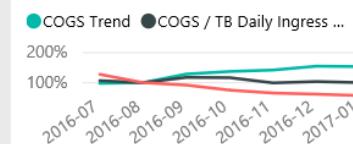
Monthly Compute COGS

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

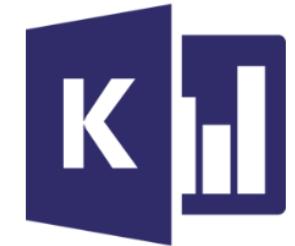
Total Queries

Total Queries

PAST WEEK • REFRESHED: THU 2:11:30 PM

117,051,200

Total Queries



Avg Daily Data Ingress (TB)

REFRESHED: FRI 5:20:46 PM

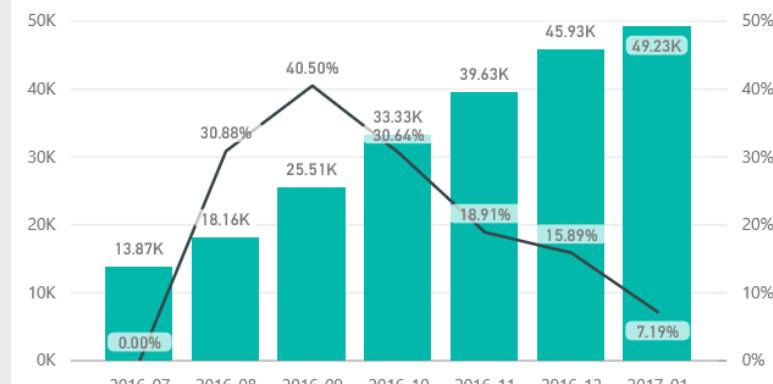
● Avg Daily Data Ingress (TB) ● MoM Avg Daily Data Ingress



Avg Total Data Size (TB)

REFRESHED: FRI 5:20:46 PM

● Avg Total Data Size (TB) ● MoM Avg Total Data Size

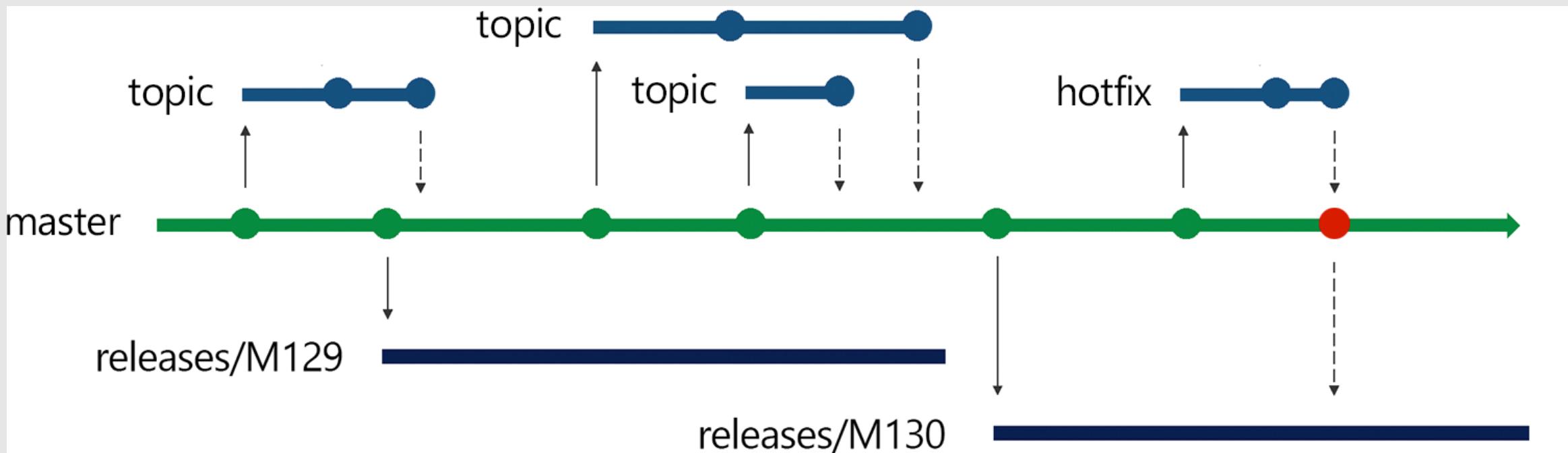


Five habits we've learned so far



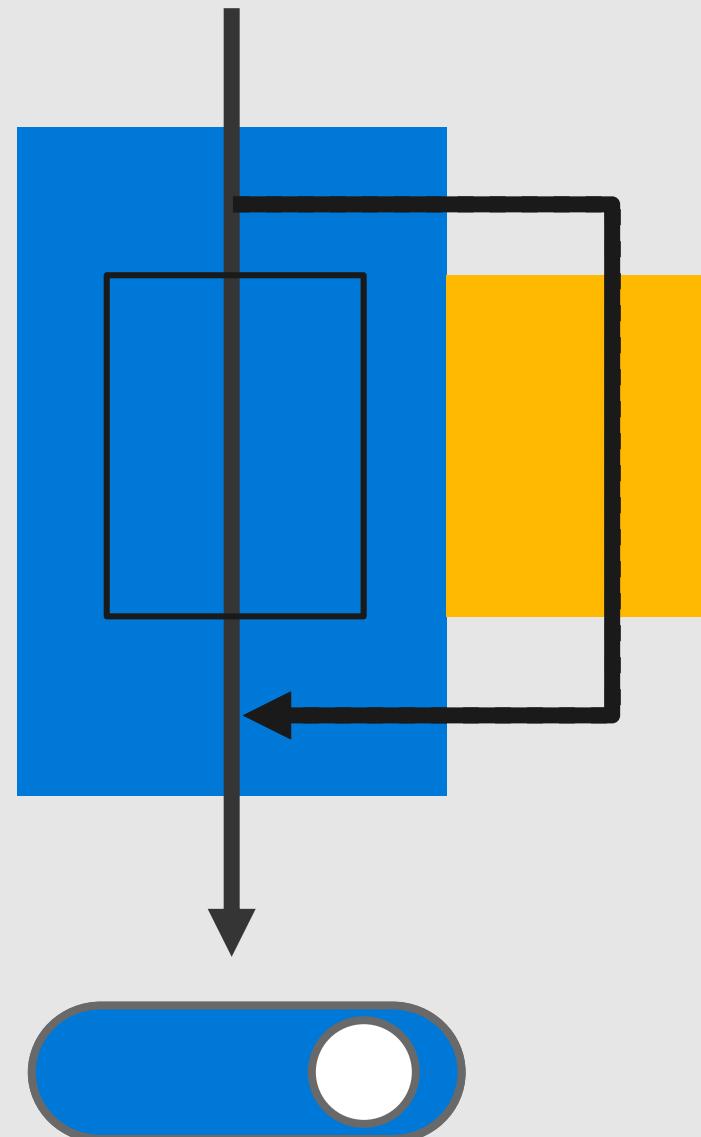
Release Flow

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



Live Site Culture

- Live site status is always the top priority
- Weekly live site review
- Root cause everything
- LSI fixes go into backlog (2 sprint rule)
- Actionable alerts
- Monthly service review
- On-call Designated Responsible Individual (DRI)
- Customer Focused Availability model (SLA)
- Per team / service health reports



Automate completely

- 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	...121.11	...121.12
Sps.SelfHost	✓ 100%								
Sps.Selftest	✓ 100%								
Tfs.ATDT	✓ 100%								
Tfs.ATTPC	✓ 100%								
Tfs.Deploy	✓ 100%								
Tfs.SelfHost	✓ 100%								
Tfs.Selftest	✓ 100%								
TfsOnPrem.SelfHost	✓ 100%								
TfsOnPrem.SelfTest	✓ 100%								

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.



Progressive Exposure

Deploy one stage at a time

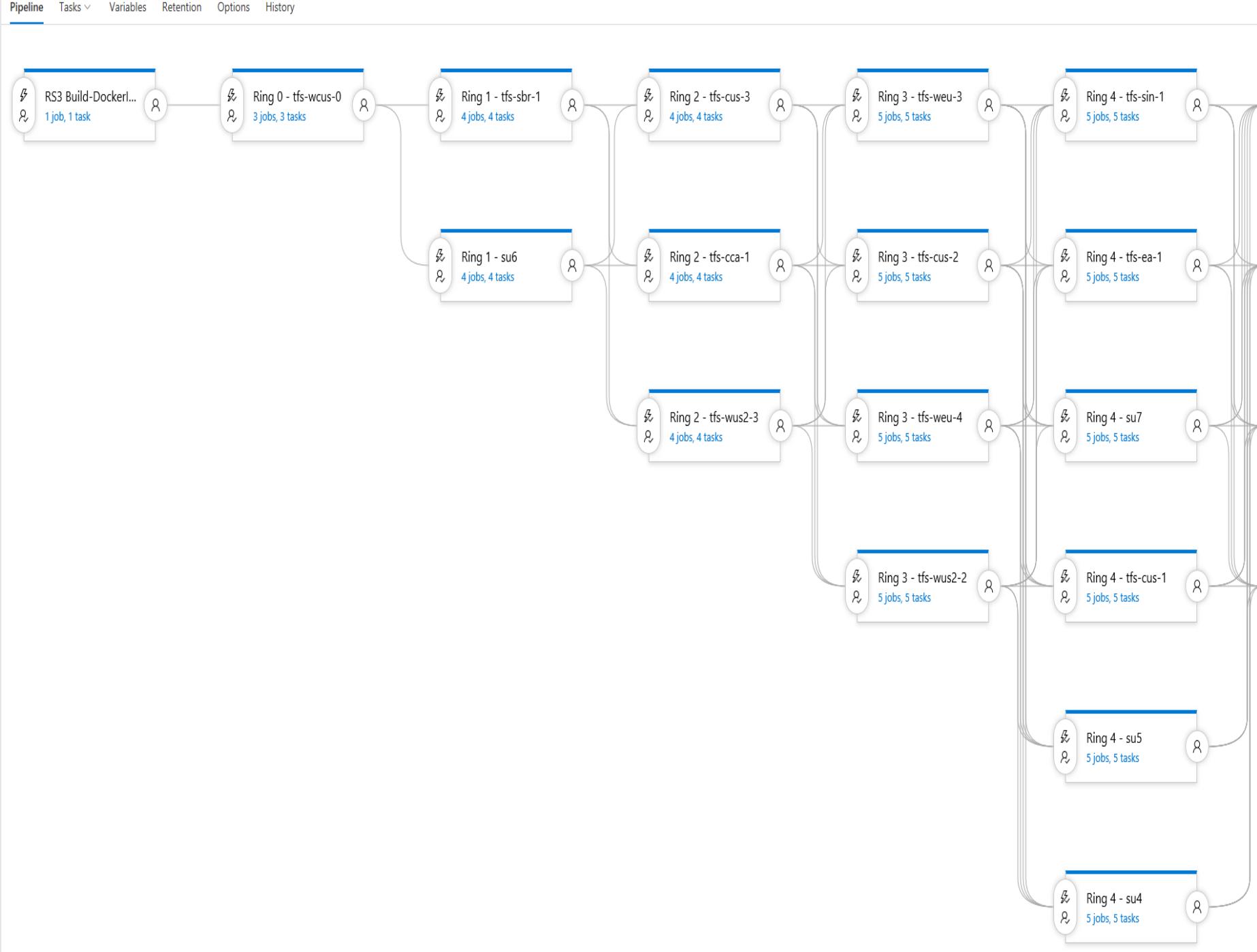
Release Gates control progression
Automated (L3) tests to check current ring
Can slow down with explicit approval

Result:

Visibility into impact of every deployment

Move across

1. Canary
2. Data centers with small user counts
3. Data center with large user count
4. Highest latency
5. The rest



Live Site Incidents

- LSI conference bridge created
- DRI's brought in to call
- Communication externally and internally
- Gather data for root cause & mitigate for customers
- Every action recorded
- Plan to rotate people during long running LSI's
- Create & track Repair Items to prevent reoccurrence and improve detection time

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 38m	TTM 40m	TTE 51m	TTN 1h 38m	Impact Duration 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.



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.

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 Service"  
| where locationName contains "weu2"  
| project PreciseTimeStamp , TaskName, status, ['title'] , description, locationName , buildNumber  
| order by PreciseTimeStamp desc
```

QoS/SLA Impact

From Clipboard

```
let startTime = datetime("2017-06-30 12:45");  
let endTime = datetime("2017-06-30 18:00");  
let service = 'tfs';  
let scaleUnit = 'tfs-weu-2';  
let commandThreshold = 1;  
let persistenceThreshold = 2;  
ActivityLog  
| where startTime >= floor(startTime, 5m) and startTime < floor(endTime, 5m)  
| where service == service  
| where scaleUnit == scaleUnit
```

Repeat Outage

No

Category
Human Error - Code Defect

Caused by Change

No

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

Five habits we've learned so far



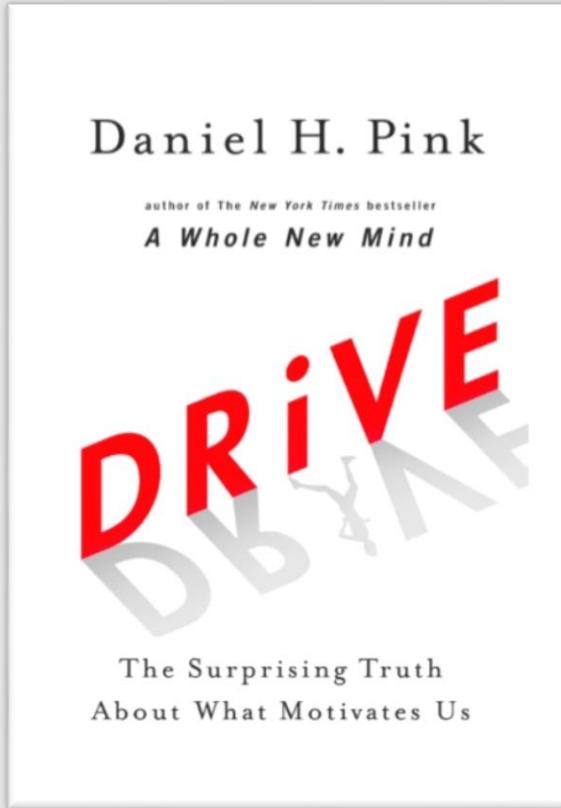
Software delivery paradox

Speed vs. control



Agile at Scale with Aligned Autonomy

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



Plan

Practices

Organization

Roles

Teams

Cadence

Taxonomy

Autonomy

Alignment

Team Structure



ORG CHART



PROGRAM
MANAGEMENT

DEVELOPMENT

TESTING

ORG CHART



PROGRAM
MANAGEMENT

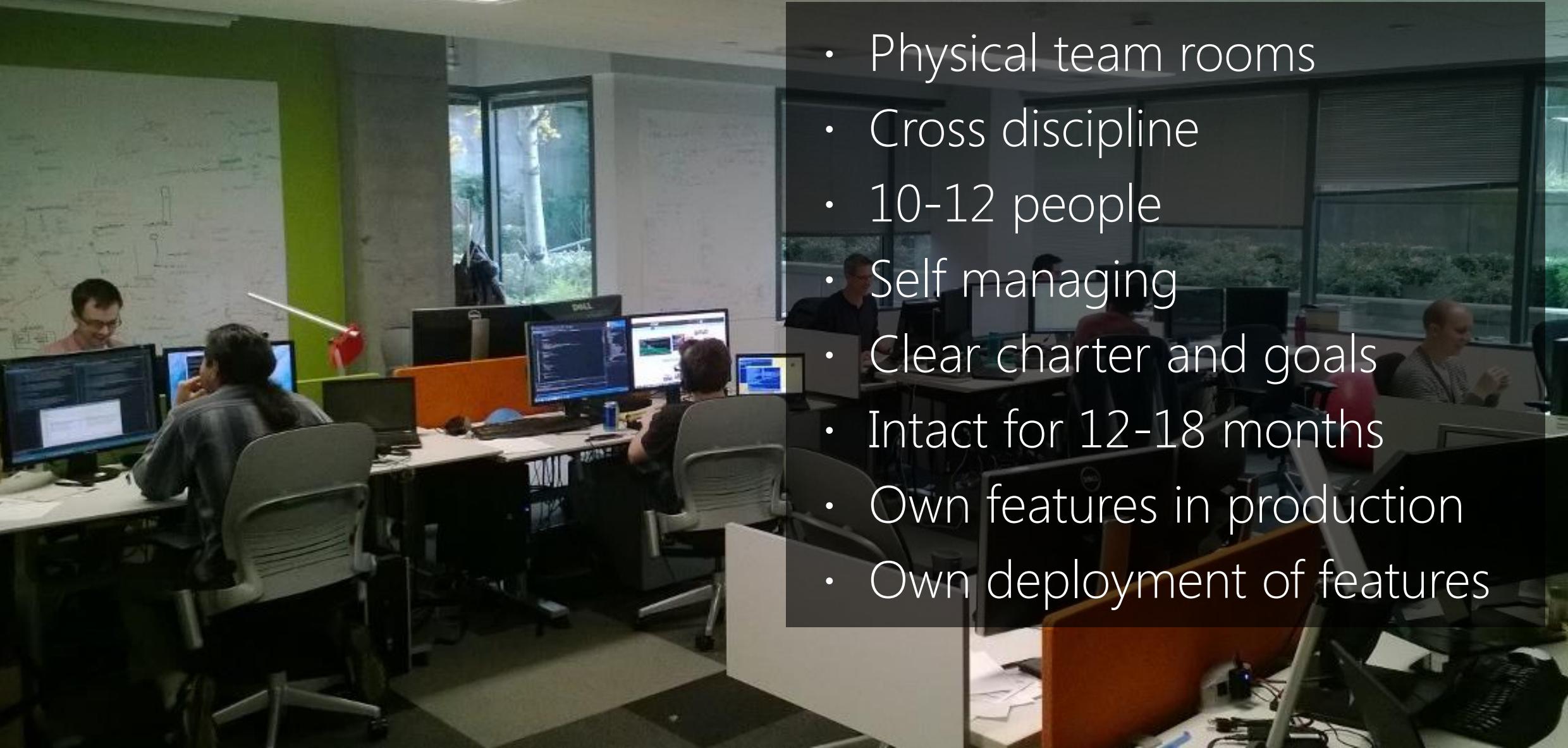


ENGINEERING



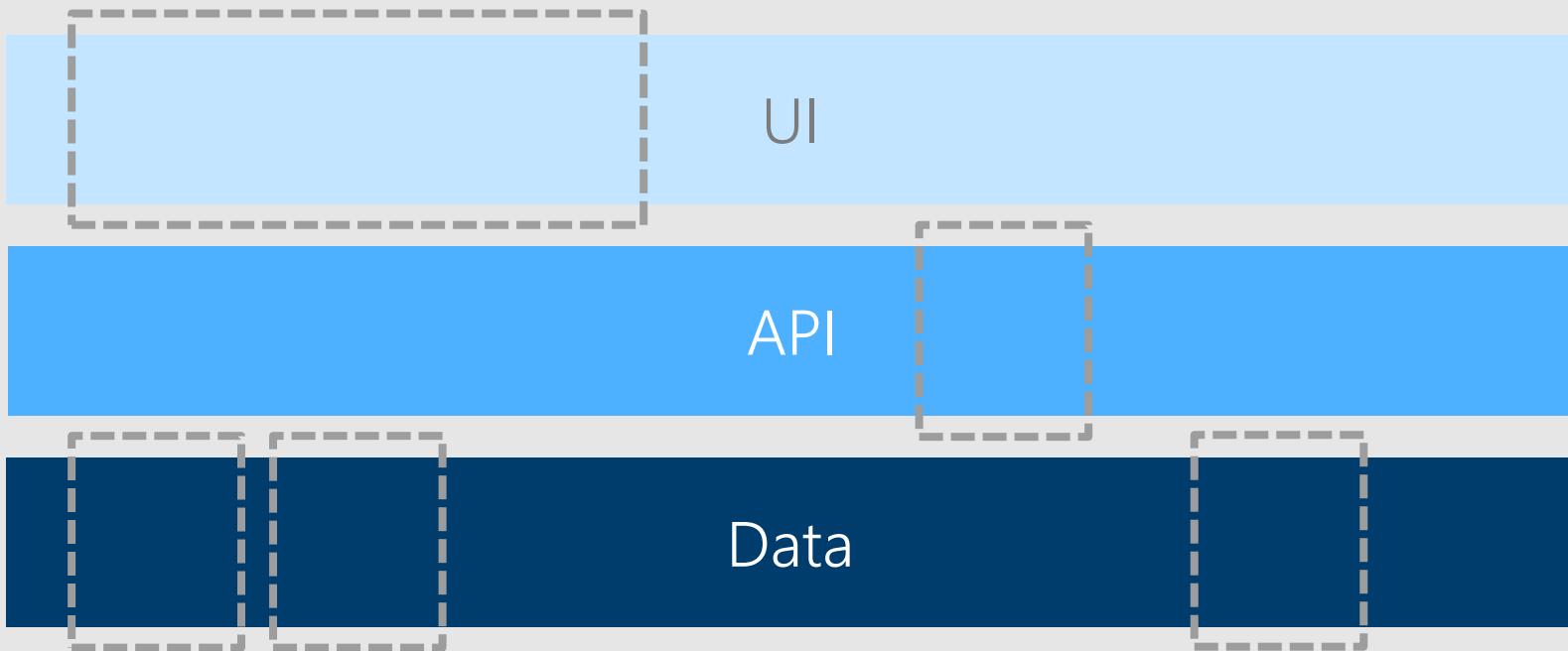
OPs

Teams

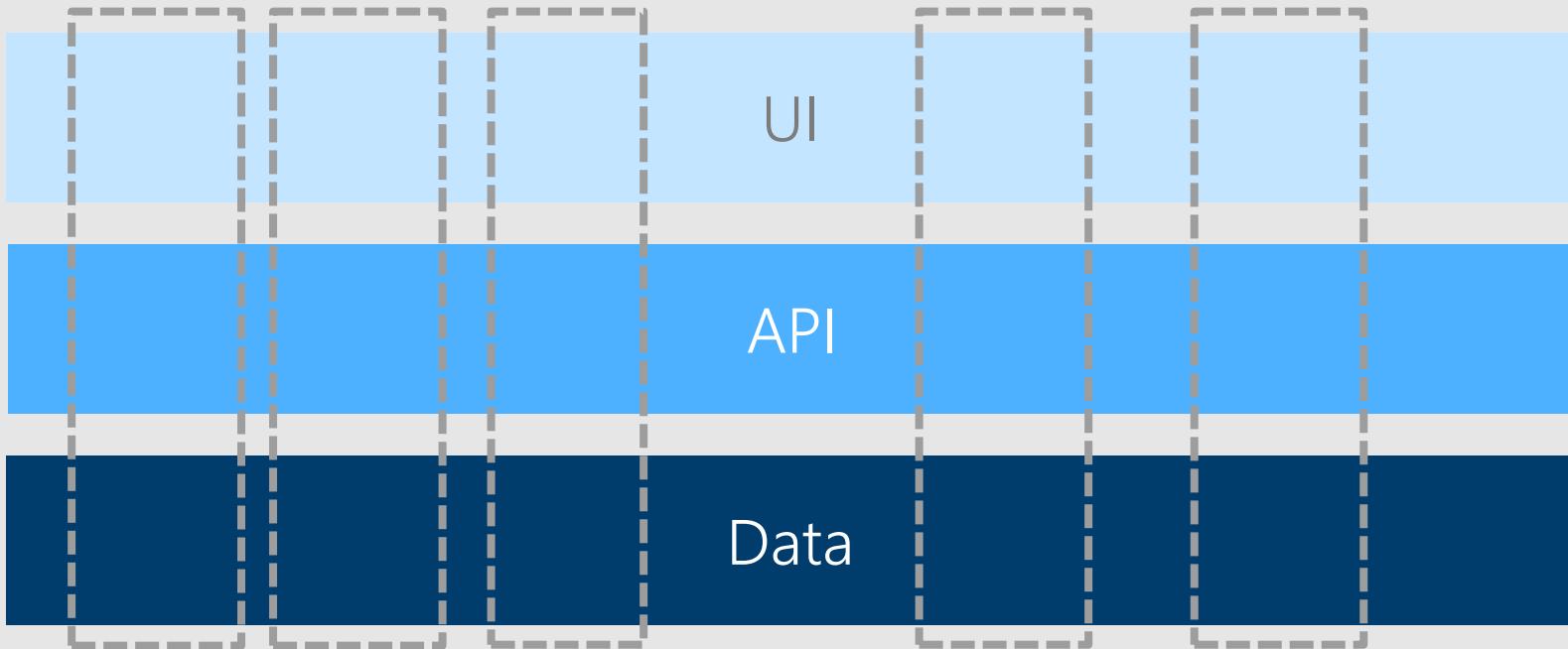


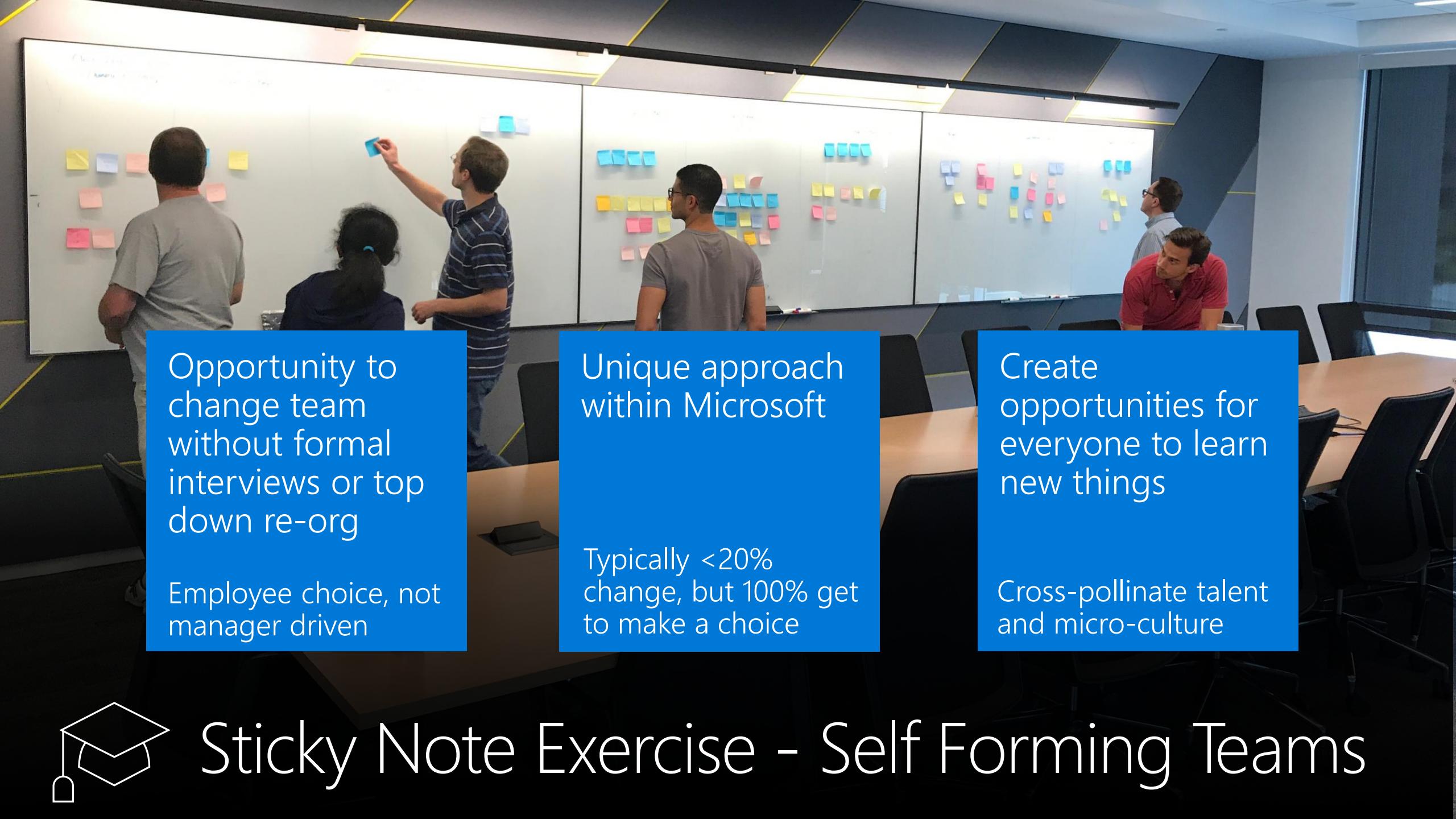
- Physical team rooms
- Cross discipline
- 10-12 people
- Self managing
- Clear charter and goals
- Intact for 12-18 months
- Own features in production
- Own deployment of features

Instead of Horizontal...



We strive for Vertical





Opportunity to change team without formal interviews or top down re-org

Employee choice, not manager driven

Unique approach within Microsoft

Typically <20% change, but 100% get to make a choice

Create opportunities for everyone to learn new things

Cross-pollinate talent and micro-culture



Sticky Note Exercise - Self Forming Teams

Planning



Leadership is responsible
for the big picture

Teams are responsible for the detail

Planning – Common Sprint Calendar

JANUARY							FEBRUARY							MARCH							APRIL									
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa			
31	1	2	3	4	5	6	Sprint 46	28	29	30	31	1	2	3	25	26	27	28	1	2	3	1	2	3	4	5	6	7		
7	8	9	10	11	12	13		4	5	6	7	8	9	10	Learning	4	5	6	7	8	9	10	8	9	10	11	12	13	14	
14	15	16	17	18	19	20	Sprint 47	11	12	13	14	15	16	17	Sprint 49	11	12	13	14	15	16	17	Jack&Learn	15	16	17	18	19	20	21
21	22	23	24	25	26	27		18	19	20	21	22	23	24		18	19	20	21	22	23	24	Sprint 51	22	23	24	25	26	27	28
28	29	30	31	1	2	3	Sprint 48	25	26	27	28	1	2	3	Sprint 50	25	26	27	28	29	30	31		29	30	1	2	3	4	5
4	5	6	7	8	9	10		4	5	6	7	8	9	10		1	2	3	4	5	6	7	6	7	8	9	10	11	12	
MAY							JUNE							JULY							AUGUST									
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa			
29	30	1	2	3	4	5		27	28	29	30	31	1	2	1	2	3	4	5	6	7	29	30	31	1	2	3	4	Sprint 59	
6	7	8	9	10	11	12	Sprint 54	3	4	5	6	7	8	9	Sprint 56	8	9	10	11	12	13	14	5	6	7	8	9	10	11	
13	14	15	16	17	18	19		10	11	12	13	14	15	16		15	16	17	18	19	20	21	Sprint 58	12	13	14	15	16	17	18
20	21	22	23	24	25	26	Sprint 55	17	18	19	20	21	22	23	Jack&Learn	22	23	24	25	26	27	28	19	20	21	22	23	24	25	
27	28	29	30	31	1	2		24	25	26	27	28	29	30	Sprint 57	29	30	31	1	2	3	4	26	27	28	29	30	31	1	
3	4	5	6	7	8	9		1	2	3	4	5	6	7		5	6	7	8	9	10	11	2	3	4	5	6	7	8	
SEPTEMBER							OCTOBER							NOVEMBER							DECEMBER									
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa			
26	27	28	29	30	31	1		30	1	2	3	4	5	6		28	29	30	31	1	2	3	25	26	27	28	29	30	1	
2	3	4	5	6	7	8		7	8	9	10	11	12	13	Jack&Learn	4	5	6	7	8	9	10	2	3	4	5	6	7	8	
9	10	11	12	13	14	15	Sprint 62	14	15	16	17	18	19	20	Sprint 64	11	12	13	14	15	16	17	9	10	11	12	13	14	15	
16	17	18	19	20	21	22		21	22	23	24	25	26	27		18	19	20	21	22	23	24	16	17	18	19	20	21	22	
23	24	25	26	27	28	29	Sprint 63	28	29	30	31	1	2	3		25	26	27	28	29	30	1	23	24	25	26	27	28	29	
30	1	2	3	4	5	6		4	5	6	7	8	9	10		2	3	4	5	6	7	8	30	31	1	2	3	4	5	

Sprint Mail – Per Epic (Current Practice)

✉ Value delivered against each Objective *for Epic*

✉ Video by *Epic* objective, not org structure

✉ Value planned for *Epic* in next sprint

Tue 11/27/2018 15:15
Aaron Bjork
Azure Boards S144/145
To Azure DevOps Sprint Communication
(i) If there are problems with how this message is displayed, click here to view it in a web browser.

[Kanban for Outlook](#) + Get more add-

Below is an update on the Boards Q4 OKR progress, <http://aka.ms/AzureBoardsOKRs>. To view the backlog of each Boards team visit <http://aka.ms/AzureBoardsFeatures>.

1. Objective: Azure Boards work items can be linked to GitHub commits and PRs.

S144: At Connect, we're releasing a public preview of the Azure Boards integration with GitHub, enabling GitHub.com users to:

1. Configure connections from a Boards project to one or more GitHub repos. (Admins only)
2. Create links by mentioning a Boards work item in a GitHub commit message, PR title, or PR description.
3. See GitHub links in the Development section in the Boards work item.
4. See Boards work items listed in the Pipelines build results when building GitHub commits that link to Boards work items.

This sprint, the admin UX was redesigned. Users studies were encouraging with users finding the key tasks of creating connections and adding repos to be straightforward. The team also completed the integration of work items into the Pipelines build summary. Updates to the [Boards marketing page](#) (to include a section that highlights the GitHub integration value-prop) were sent to the ACOM team ([designs](#)). A blog post for the Azure Blog has been drafted and is out for review. Content plans have also been prepared to help users configure and use the integration.

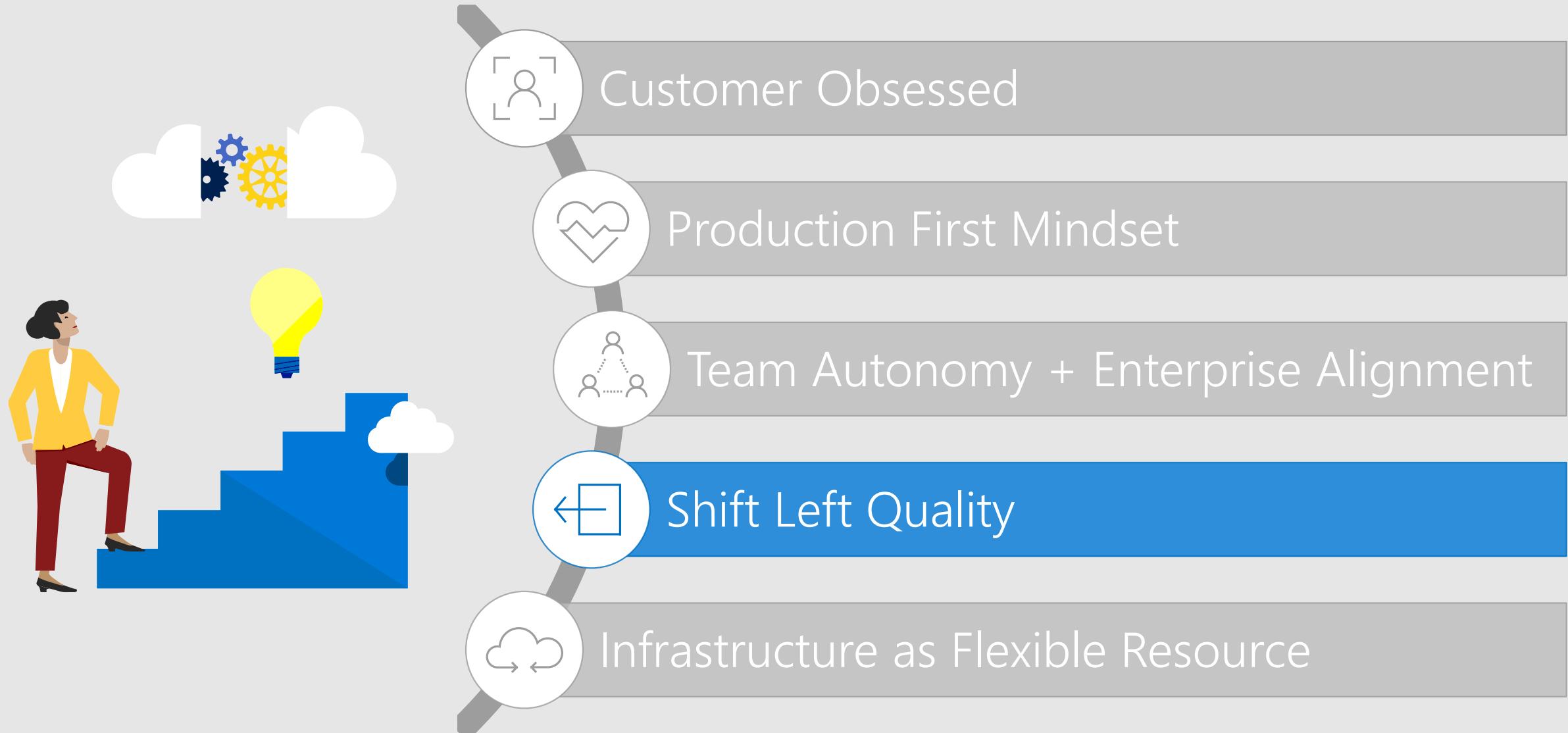
S145: The first half of Sprint 145 will focus on preparation for Connect. We'll start the sprint with a UX walkthrough to identify and prioritize any items that need to be hot fixed. Testing and daily walkthroughs will continue up to launch. All content supporting the launch (marketing page, blog post, release notes, docs) will be finalized and published.

After launch, we'll be focusing on landing GitHub Enterprise support for Azure DevOps Server RC2.

2. Objective: Azure Boards is the leading choice for teams using GitHub.

S144: With this being the last sprint before Connect() on December 4th, we shifted priorities mid-sprint to immediately address bumps and roadblocks in our acquisition funnel and day-1 experiences.

Five habits we've learned so far



Created a “Non-Denominational” Test Taxonomy

L0

In-memory unit tests with no dependencies

In Pull Request
< 60 ms

L1

Unit tests with dependencies e.g. SQL

< 2000 ms
(avg 400 ms)

L2

Functional tests against APIs

In pre-prod deployment, with mocked auth

L3

PROD

In production rings

Testing: Shift Left from Integration to Unit

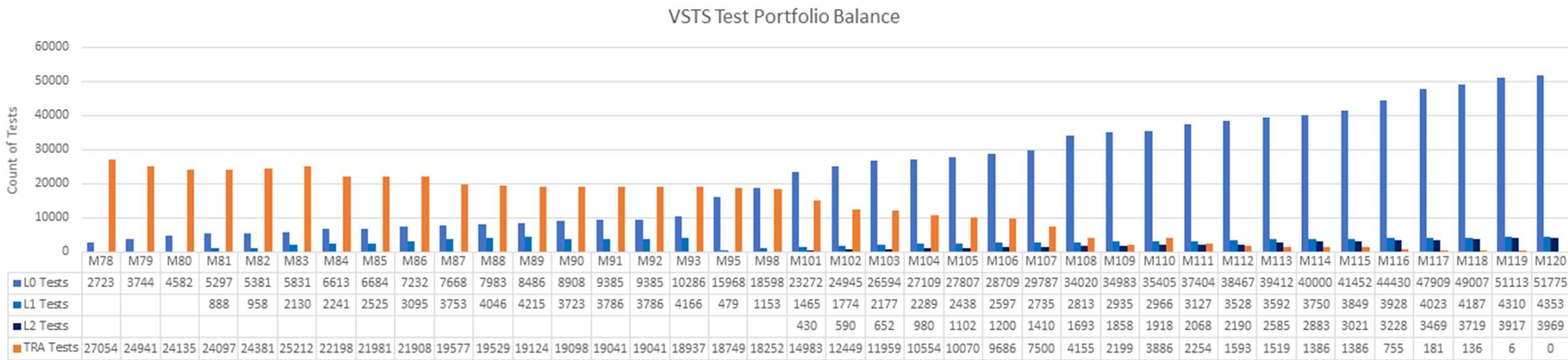
L0 – Requires only built binaries, no dependencies

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



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

The screenshot shows the Azure DevOps interface for a pull request. The URL is https://dev.azure.com/mseng/_git/AzureDevOps/pullrequest/260803. The pull request is labeled 'COMPLETED' and has the title 'Added keyboard shortcuts to Queries pivot page #1081972'. It was merged by Karthik Balasubramanian on 23/09/2017 at 18:45, with a squash merge. The PR was merged into the 'master' branch from 'users/kabalas/addshortcuts'. The 'Overview' tab is selected, showing links for 'Files', 'Updates', 'Commits', 'Additional Validations', and 'Conflicts'. On the right, there are sections for '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' (Bug 1081972: Add shortcuts to query directory page), and 'Reviewers' (WIT IQ via Matthew Manela, Matthew Manela Approved). The main pane displays the pull request history with comments from Karthik and Matthew Manela.

260803 COMPLETED Added keyboard shortcuts to Queries pivot page #1081972

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

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 @Karthik Balasubramanian 🤗 I approved but make sure it works after XHR navigate since we hit issues with that before Resolved ▾

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
 - ✓ ChangelImpactAnalysis
 - ✓ 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

Edit build definition Queue new build... Download all logs as zip Retain indefinitely Release

Build succeeded

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

Summary Timeline Artifacts Code coverage* Tests WhiteSource Bolt Build Report

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 debsha
Adding Public access 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)	0 (+0)	100% (+100%)	20m 7s (+20m 7s)

Passed (78104) Failed (0) New (0) Existing (0) Others (0)

Not Reported 295

[Detailed report >](#)

Code Coverage

No build code coverage data available.

Feedback in minutes, before acceptance of PR

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%								
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%

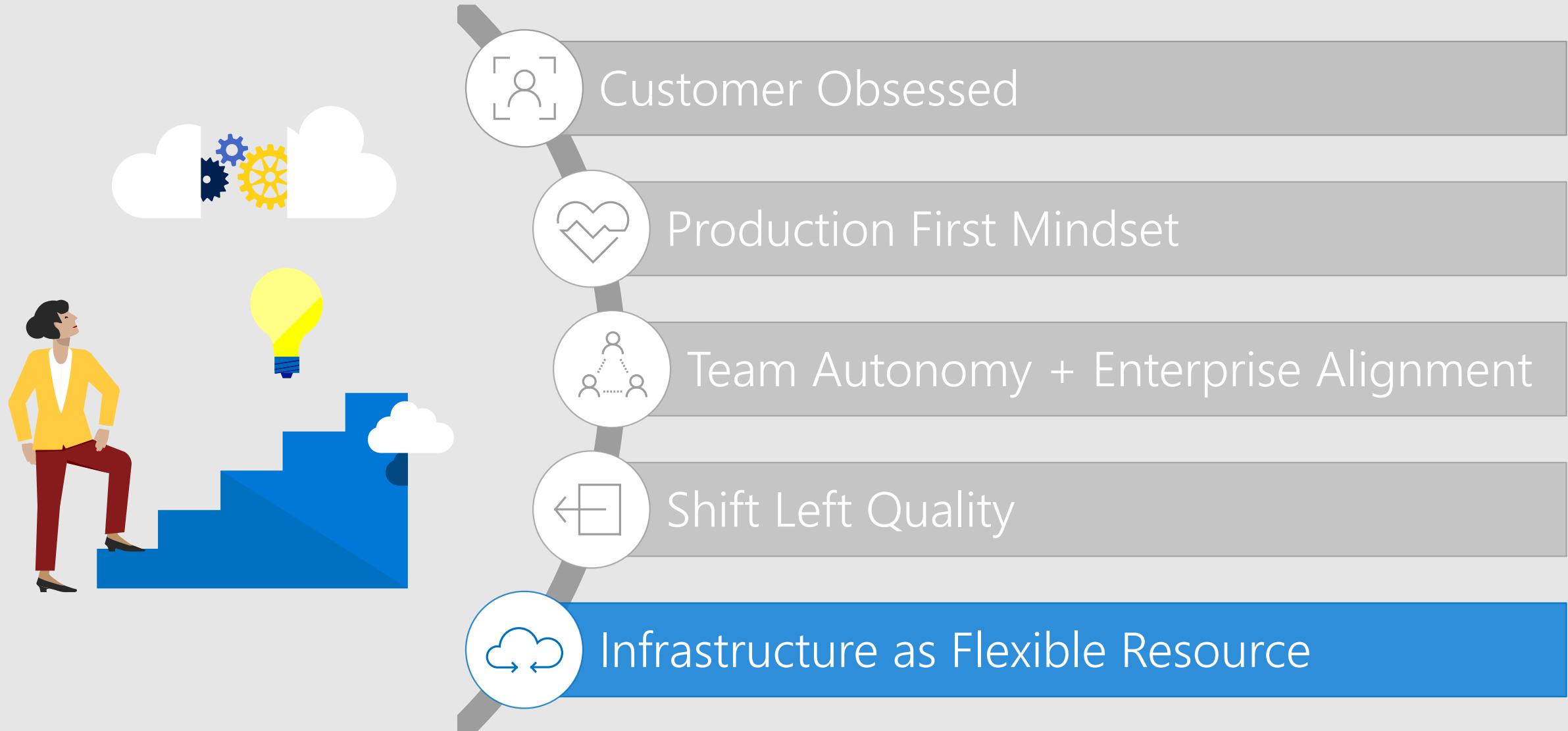
Only all-green builds get to release

Do Not Incur Debt

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

$$\# \text{ engineers on your team} \times 5 = ?$$

Five habits we've learned so far

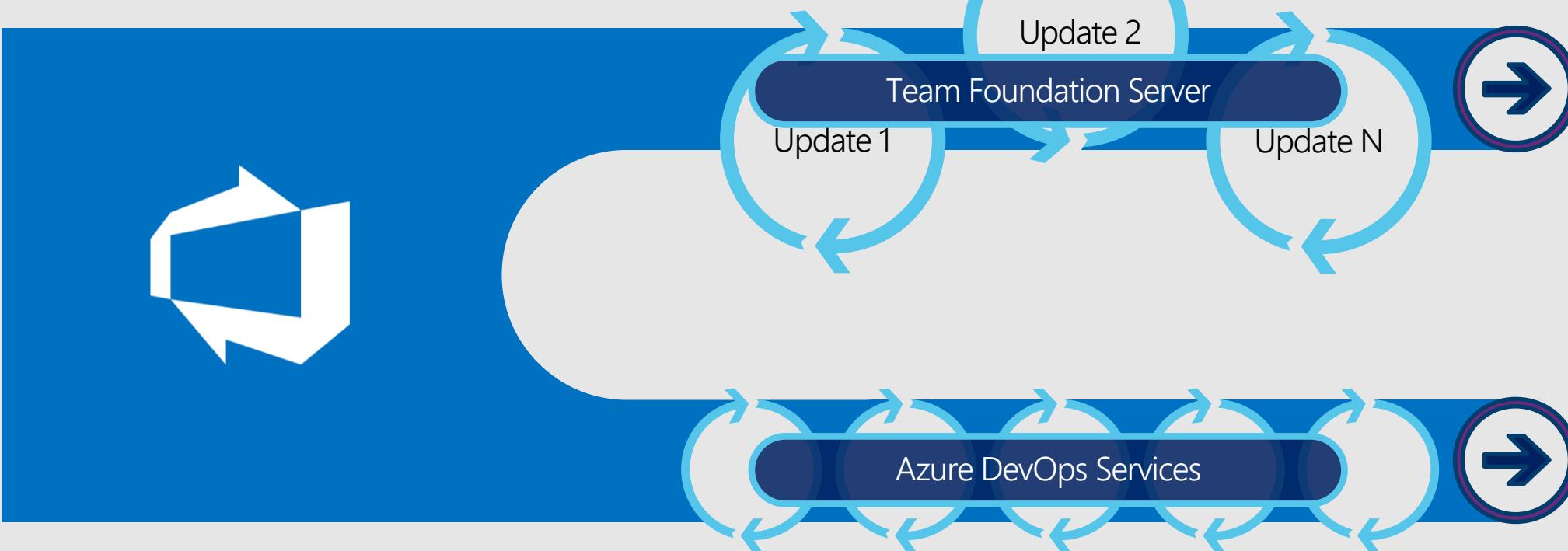


Code: Cloud first, then move on-premises

One code base with multiple delivery streams

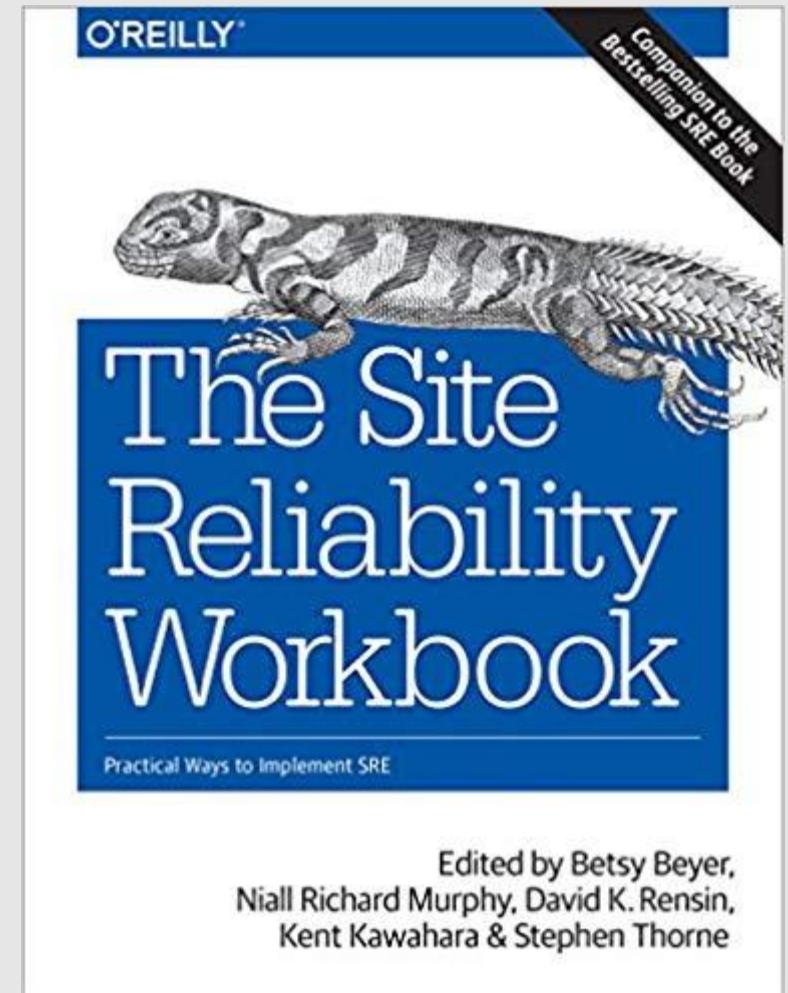
Shared abstraction layer

Single master branch, multiple release branches



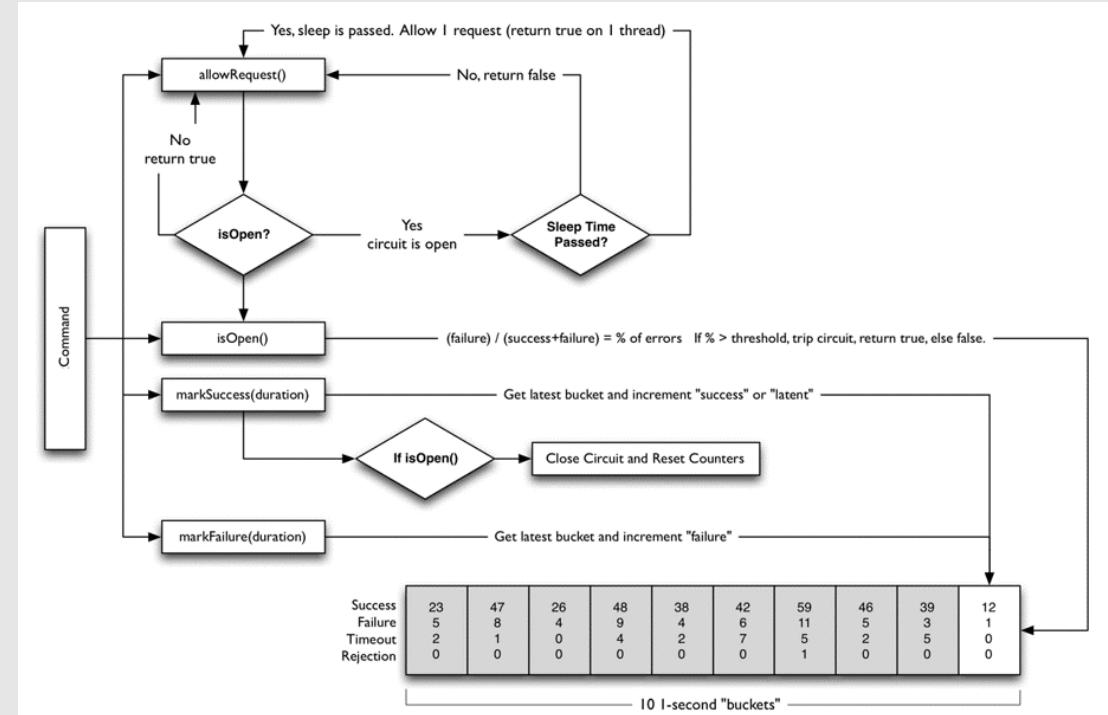
Building the SRE Discipline

- 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



Resilience Engineering

- Design with failure in mind
- Circuit breakers
- Self healing systems
- Safely introducing faults to test resilience
- Goal: Chaos engineering by default across Azure services



Our DevOps Transformation – the story so far

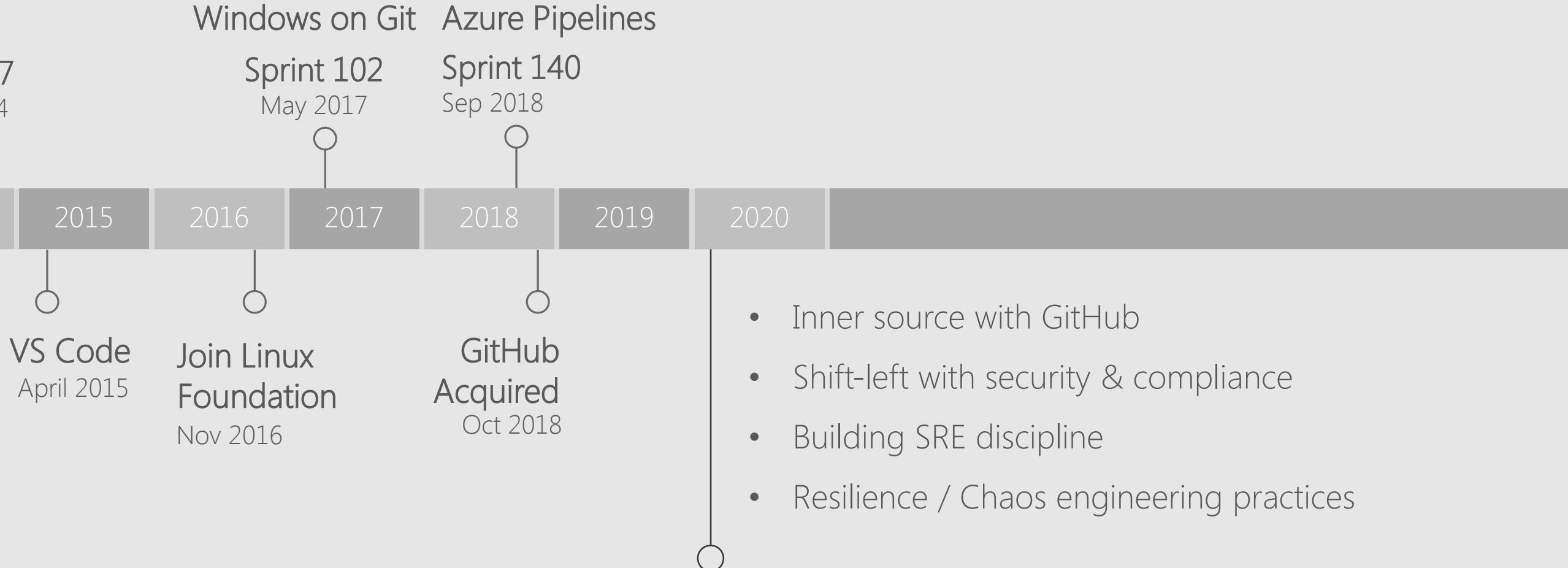
Before

- 4-6 month milestones
- Horizontal teams
- Personal offices
- Long planning cycles
- PM, Dev, Test
- Yearly customer engagement
- Feature branches
- 20+ person teams
- Secret roadmap
- Bug debt accumulated
- 100 page spec documents
- Private repositories
- Deep organizational hierarchy
- Success is a measure of install numbers
- Features shipped once a year

After

- 3-week sprints
- Vertical teams
- Team rooms
- Continual Planning & Learning
- PM & Engineering
- Continual customer engagement
- Everyone in master
- 8-12 person teams
- Publicly shared roadmap
- Debt paid as incurred
- Mockups in PPT
- Inner source
- Flattened organization hierarchy
- User satisfaction determines success
- Features shipped every sprint

What's next?





Thank you!

<http://aka.ms/MSDevOps>

Michael Royster

mroyster@microsoft.com