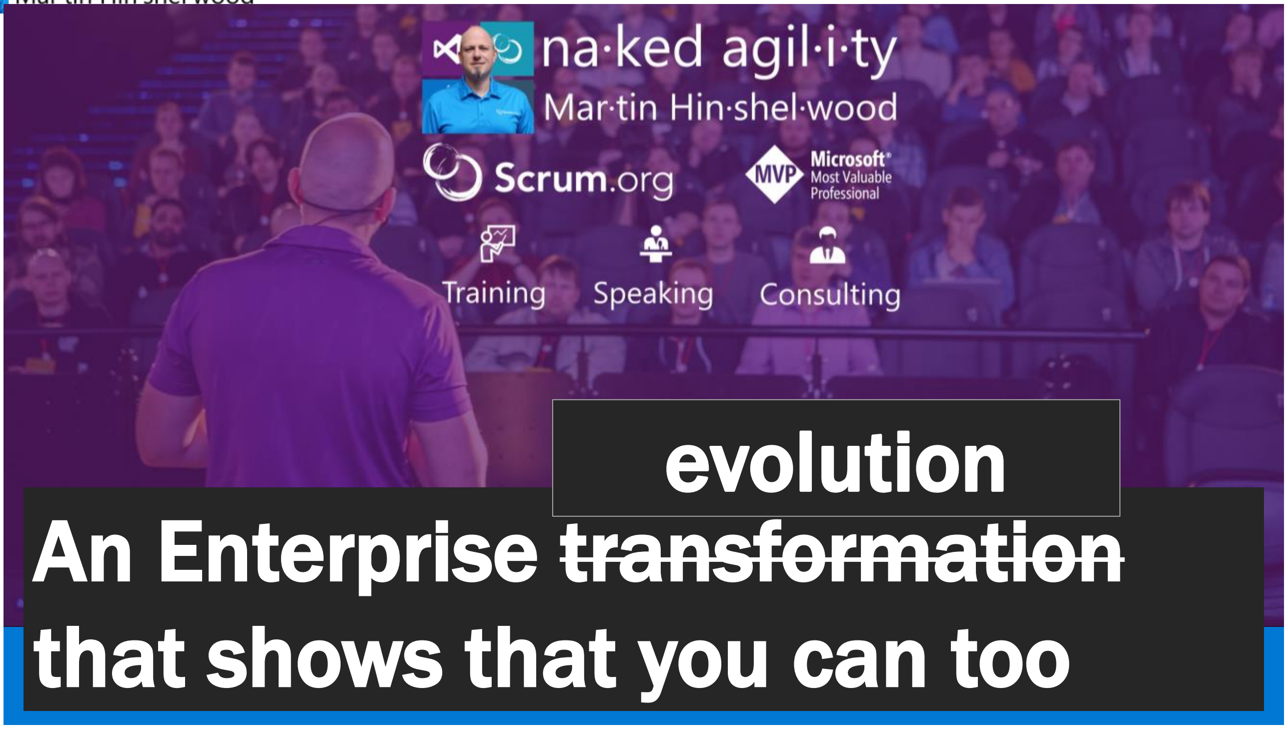


na·ked agil·i·ty
Mar·tin Hin·shel·wood

Scrum.org

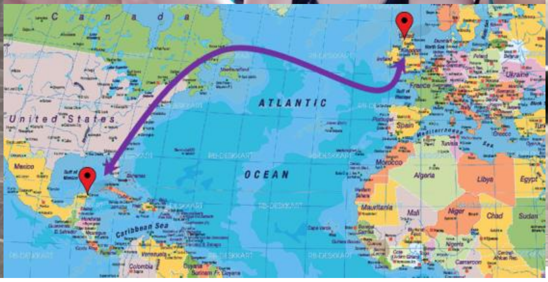
Microsoft® Most Valuable Professional MVP

Training Speaking Consulting

An Enterprise ~~transformation~~ evolution that shows that you can too

1

Martin Hinshelwood - Personal



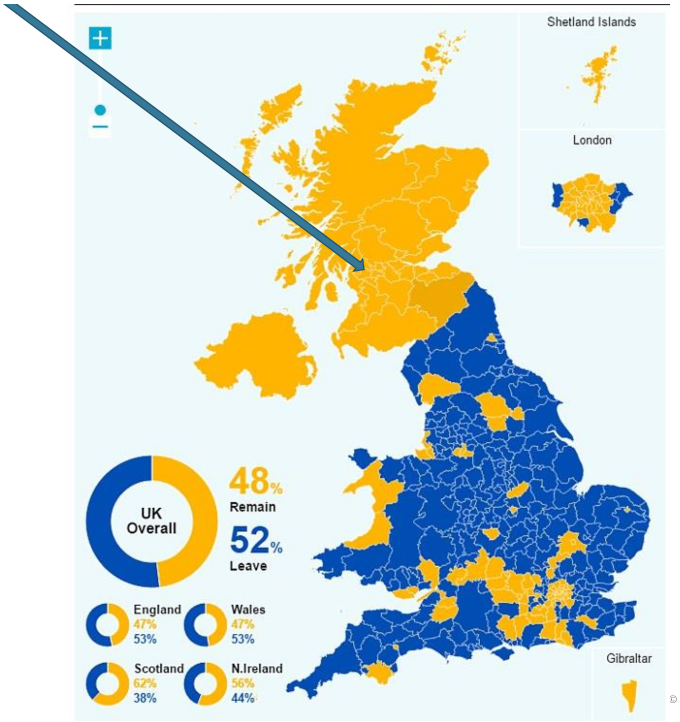
na·ked agil·i·ty
Mar·tin Hin·shel·wood

@MrHinsh

2

na·ked agil·i·ty
Mar·tin Hin·shel·wood

3



4

na·ked agil·i·ty
Mar·tin Hin·shel·wood

NDC { Oslo }
PRO VILNIUS
Professional Application Lifecycle Management with Visual Studio 2013

Martin Hinshelwood

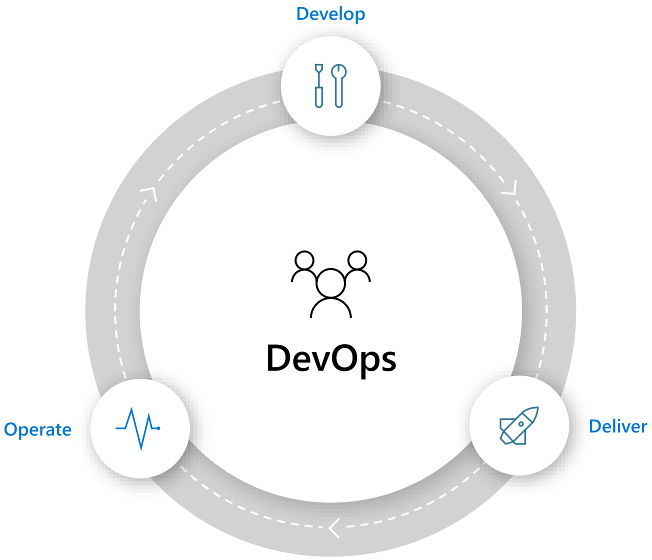
martin@nkdAgility.com

@MrHinsh

5

What is DevOps

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



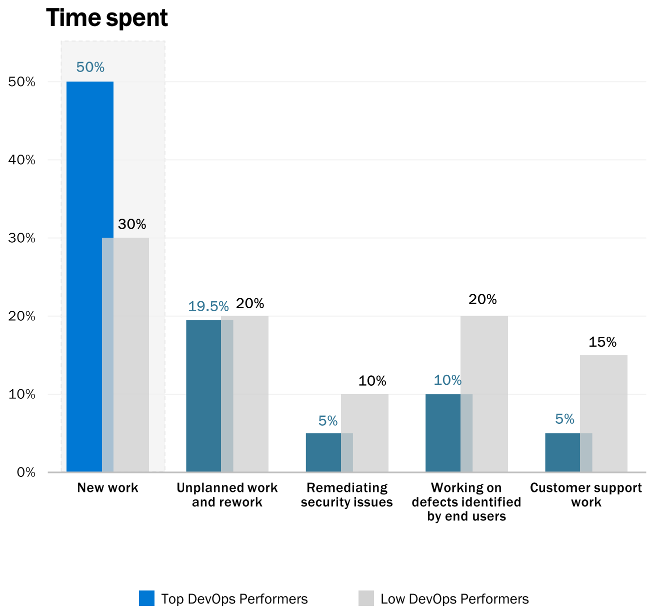
6

na·ked agil·i·ty
Mar·tin Hin·shel·wood

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



DORA DEVOPS RESEARCH & ASSESSMENT **Accelerate: State of DevOps 2018**: Strategies for a New Economy

7

Times have changed!

> “Firms today experience a much higher velocity of business change. Market opportunities appear or dissolve in months or weeks instead of years. ”

Diego Lo Giudice and Dave West, Forrester
February 2011
Transforming Application Delivery

8

na·ked agil·i·ty
Mar·tin Hin·shel·wood

Can you think of any epic failures?

Poor Quality	Mismatch to customer desires

na·ked agil·i·ty Mar·tin Hin·shel·wood @MrHinsh

9

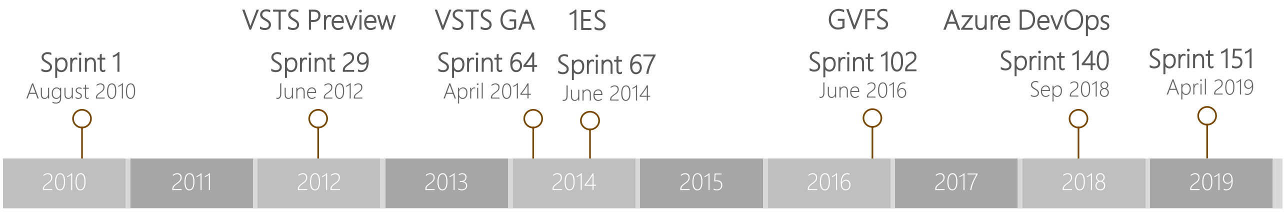
This is the story of:

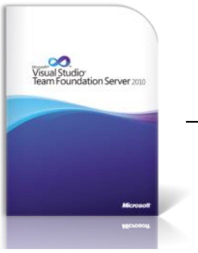
na·ked agil·i·ty Mar·tin Hin·shel·wood @MrHinsh

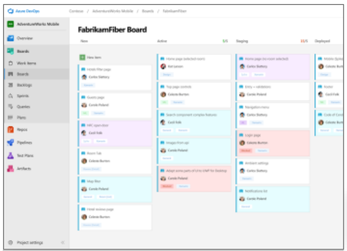
10

na·ked agil·i·ty
Mar·tin Hin·shel·wood

Journey to DevOps







Microsoft
na·ked agil·i·ty
Mar·tin Hin·shel·wood
@MrHinsh

11

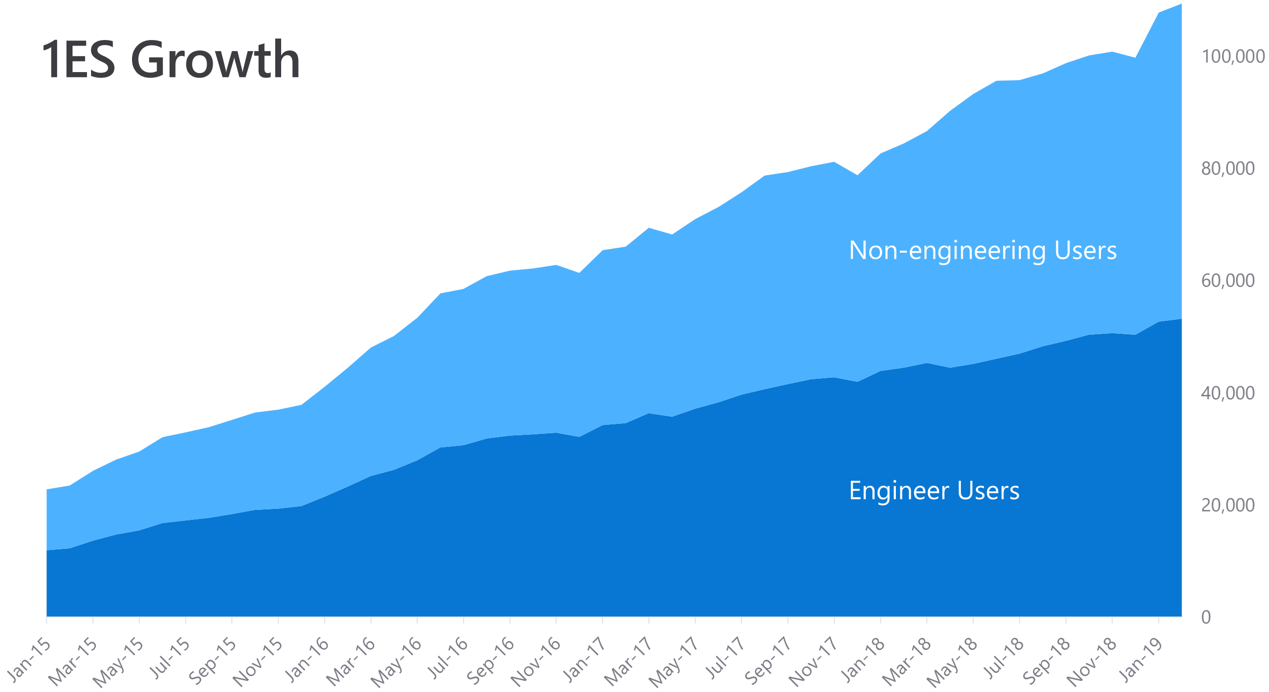
One Engineering System using Azure DevOps

> 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

12

na·ked agil·i·ty
Mar·tin Hin·shel·wood

1ES Growth



13

DevOps at Microsoft

Azure DevOps is the toolchain of choice for Microsoft engineering with over 100,000 internal users

https://aka.ms/DevOpsAtMicrosoft

442k Pull Requests per month

155 Petabytes of build artifacts managed

28k Work items created per day

2.4m Private Git commits per month

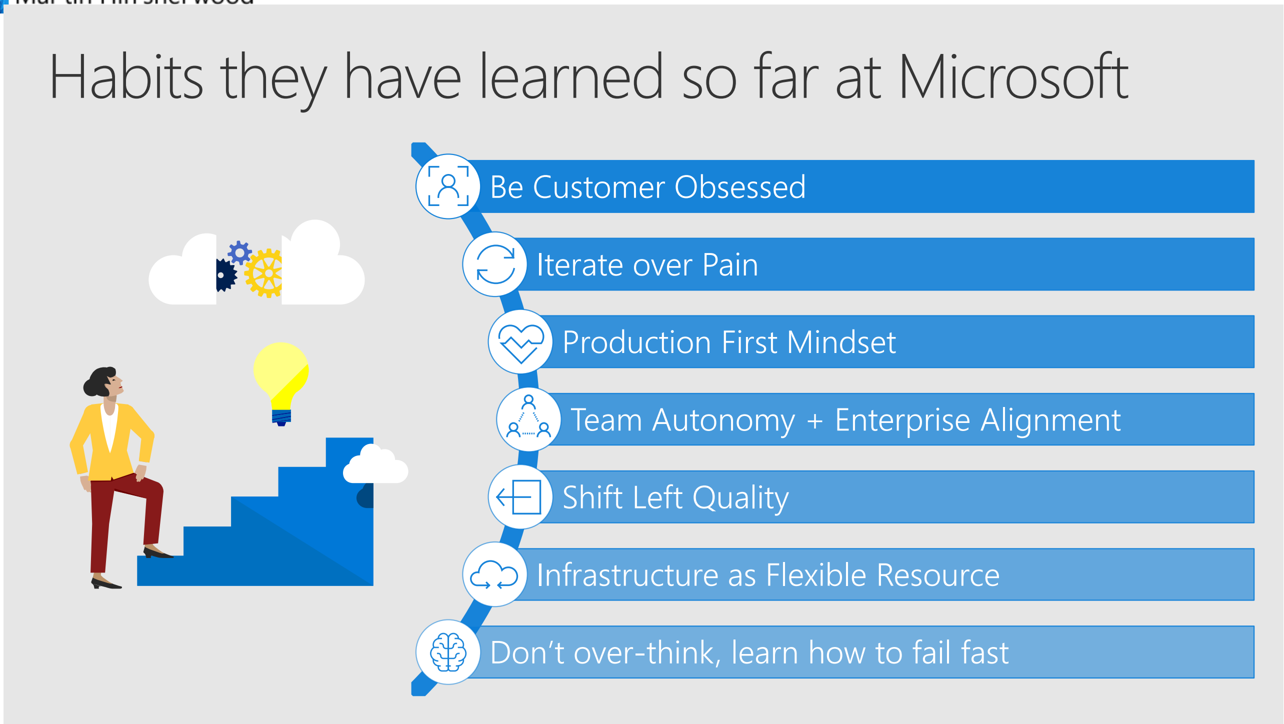
3.5k Open Source repos

12k Employees contributing to open source

82,000 Deployments per day

Data: Internal Microsoft engineering system activity, March 2019

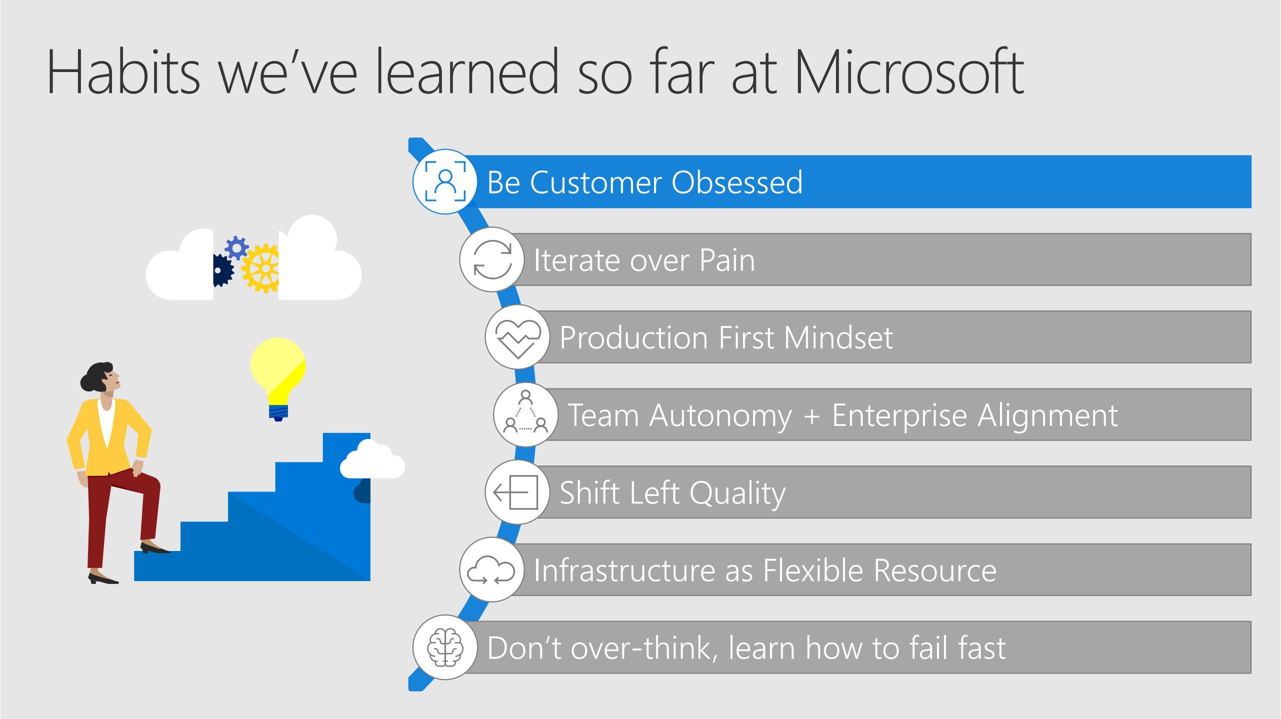
14



Habits they have 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

15



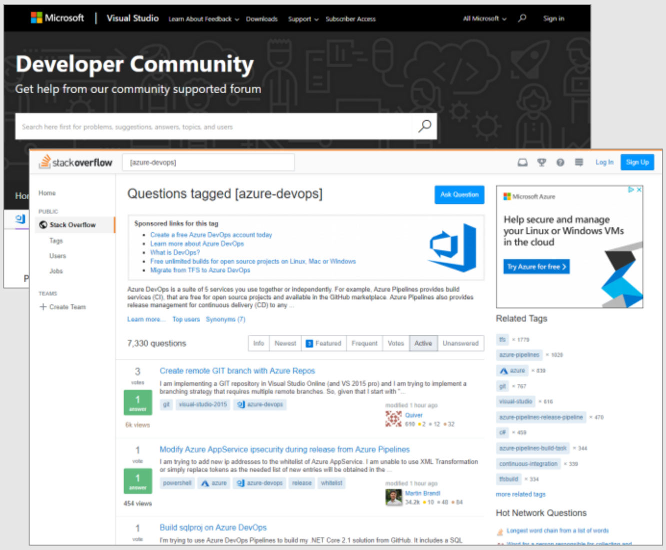
Habits we’ve learned so far at Microsoft

- 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

16

Listen to their 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 >

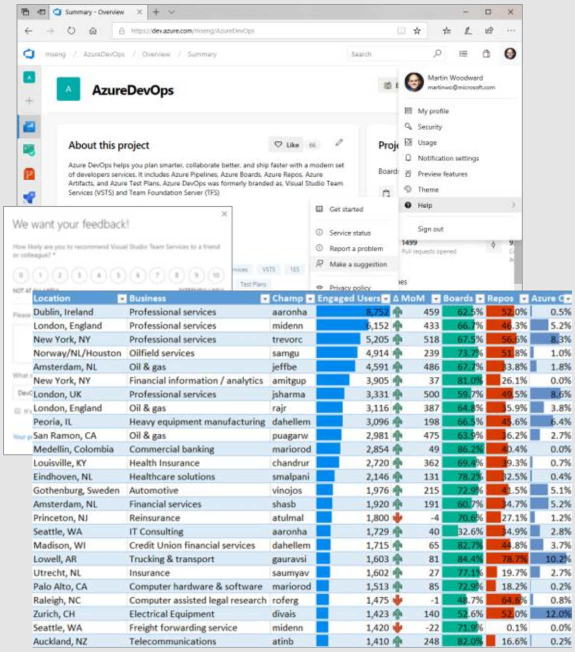
Related Tags

tfs	× 1779
azure-pipelines	× 1028
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



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

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

NOT AT ALL LIKELY EXTREMELY LIKELY

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%

17

Their Definition of Done

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

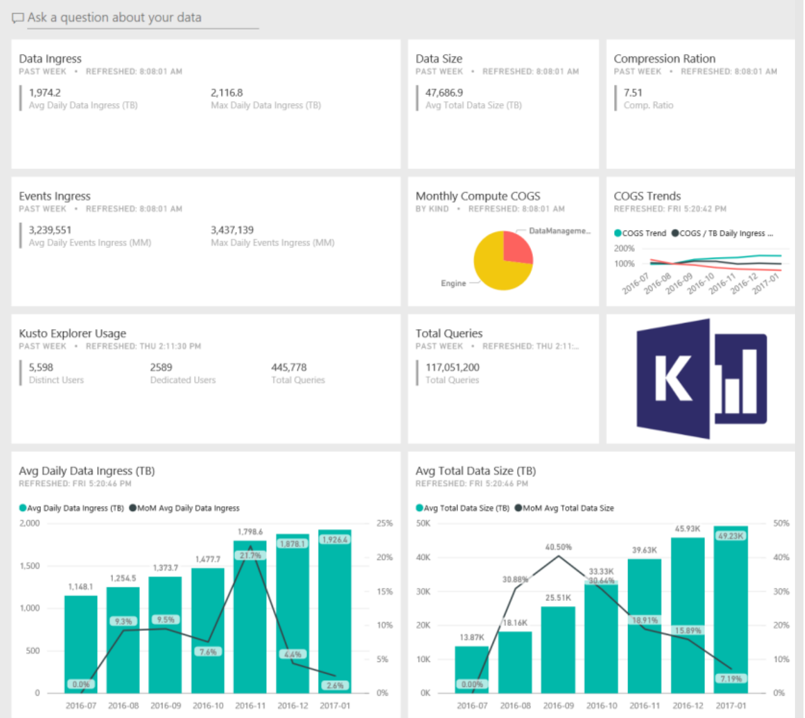
18

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

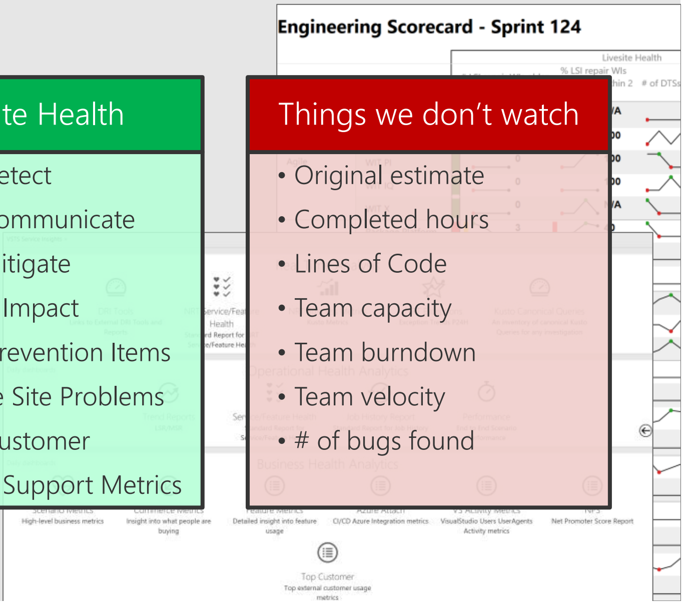
Total Queries
PAST WEEK • REFRESHED: THU 2:11:...
117,051,200 Total Queries

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

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

19

But measure what's important (KPI's)



Engineering Scorecard - Sprint 124
Livesite Health
% LSI repair WIs
# of DTSs
Top Customer
Top external customer usage metrics
High-level business metrics
Insight into what people are buying
Detailed insight into feature usage
CI/CD Azure Integration metrics
VisualStudio Users UserAgents Activity metrics
Net Promoter Score Report

Usage

- Acquisition
- Engagement
- Satisfaction
- Churn
- Feature Usage

Velocity

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

Live Site Health

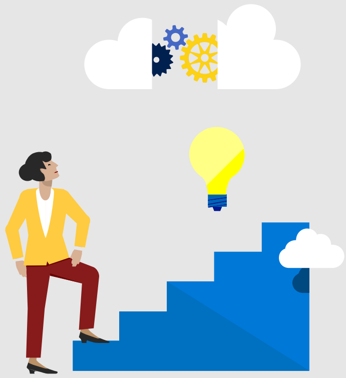
- Time to Detect
- Time to Communicate
- Time to Mitigate
- Customer Impact
- Incident Prevention Items
- Aging Live Site Problems
- SLA per Customer
- Customer Support Metrics

Things we don't watch

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

20

Habits they have learned so far at Microsoft



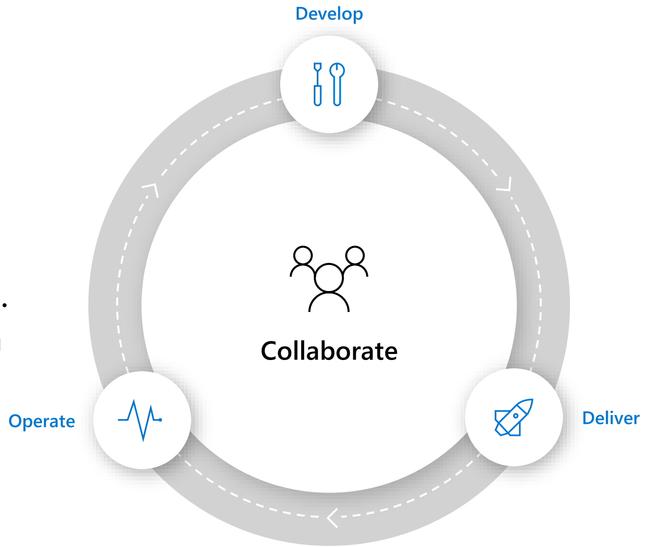
- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don't over-think, learn how to fail fast

21

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.

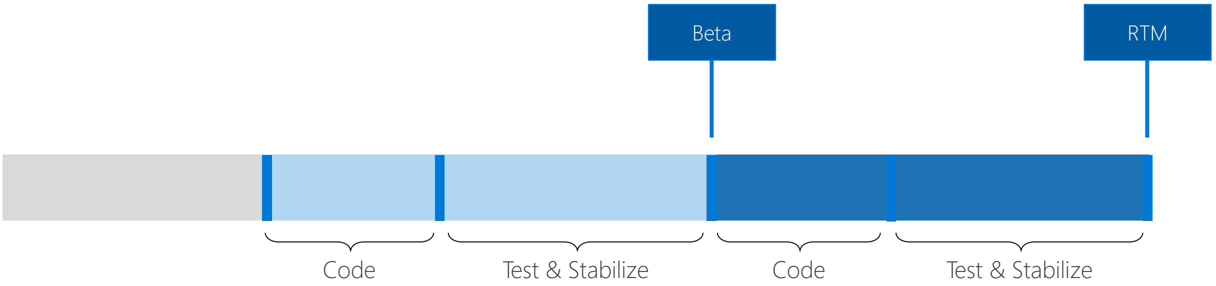


22

na·ked agil·i·ty
Mar·tin Hin·shel·wood

Schedule

The OLD way



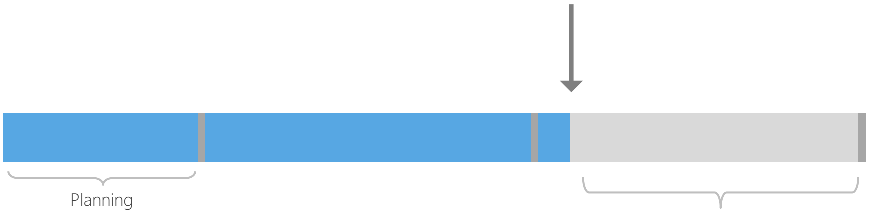
na·ked agil·i·ty Mar·tin Hin·shel·wood @MrHinsh

23

Feedback

The OLD way

Customer feedback – we should change the way a feature works. We didn't get it *quite* right…



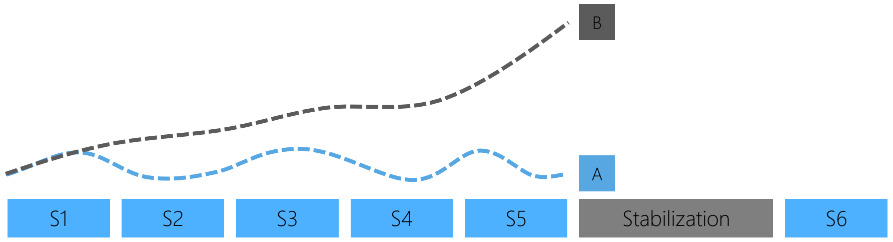
… but we're booked solid already.

na·ked agil·i·ty Mar·tin Hin·shel·wood @MrHinsh

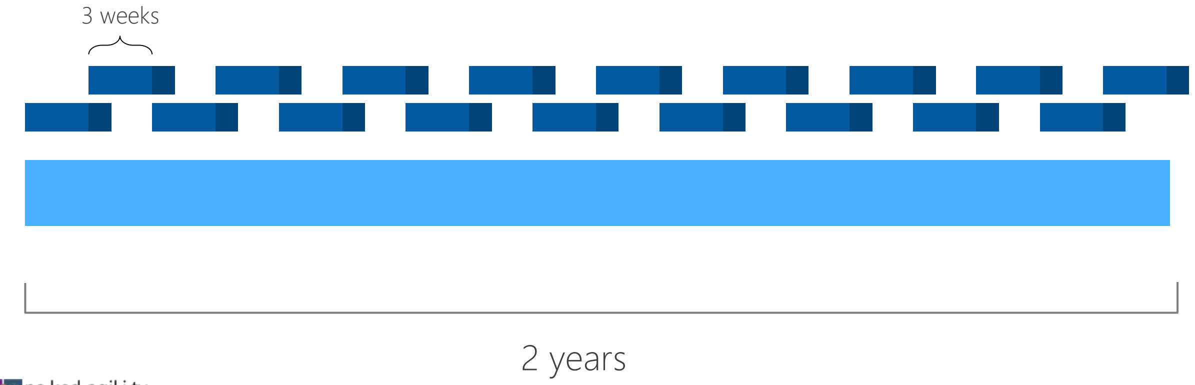
24

Story: Sprint 1-5

Take Two



Now



na·ked agil·i·ty
Mar·tin Hin·shel·wood

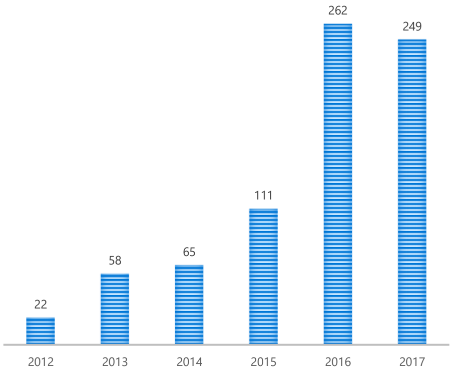
Features Delivered per Year

Deliver more value to customers

Faster responses to customers and market changes

Improved engineering satisfaction

2x productivity increase



na·ked agil·i·ty
Mar·tin Hin·shel·wood

@MrHinsh

https://www.visualstudio.com/en-us/articles/news/features-timelin

27

Maintaining enterprise rigor

Everyone is on ONE main master branch

Git helps with lightweight topic branching

Tiny, continuous merging

Code is fresh in your mind

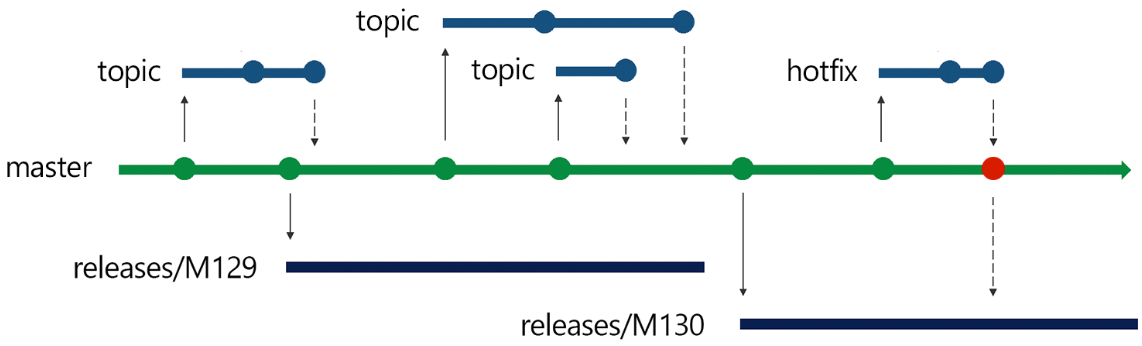
28

na·ked agil·i·ty
Mar·tin Hin·shel·wood

Release Flow

Using Trunk Based Development to avoid Merge Hell

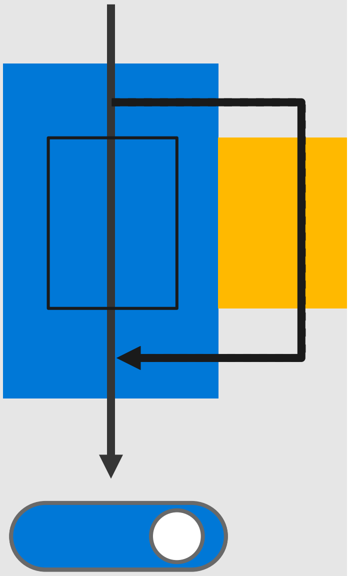
The NEW way



29

Feature Flags

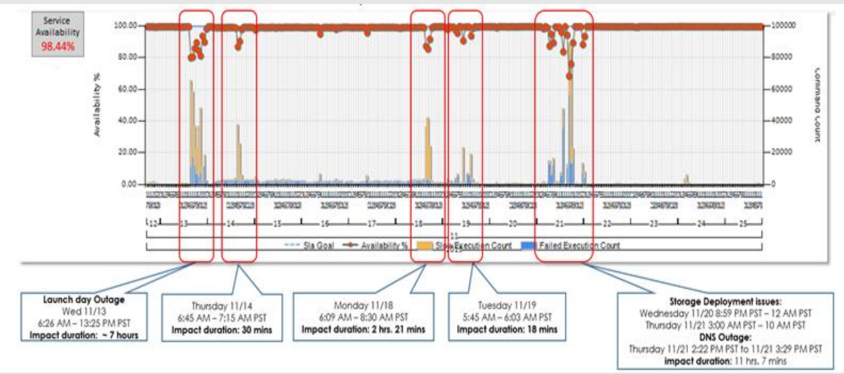
- All code is deployed, but feature flags control exposure
  - Reduces integration debt
- Flags provide runtime control down to individual user
- Users can be added or removed with no redeployment
- Mechanism for progressive experimentation & refinement
- Enables dark launch



30

Awesome! What could go wrong?

- Features to be revealed at big event
- We turned features on globally just before the keynote…
- It didn't go well.



Launch day Outage
Wed 11/13
6:26 AM – 13:25 PM PST
Impact duration: ~ 7 hours

Thursday 11/14
6:45 AM – 7:15 AM PST
Impact duration: 30 mins

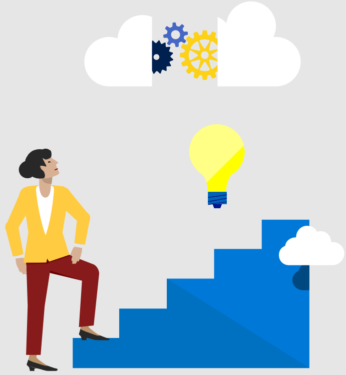
Monday 11/18
6:09 AM – 8:30 AM PST
Impact duration: 2 hrs. 21 mins

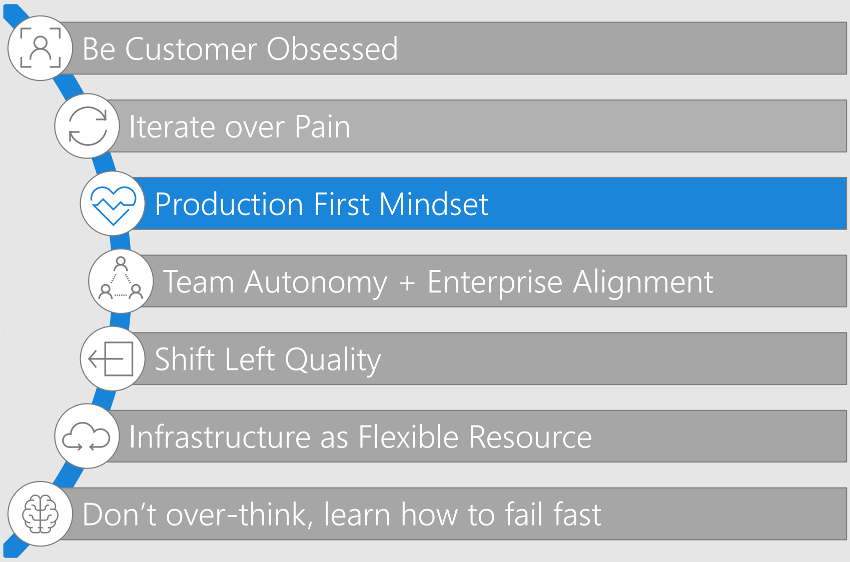
Tuesday 11/19
5:45 AM – 6:03 AM PST
Impact duration: 18 mins

Storage Deployment issues:
Wednesday 11/20 8:59 PM PST – 12 AM PST
Thursday 11/21 3:00 AM PST – 10 AM PST
DNS Outage:
Thursday 11/21 2:22 PM PST to 11/21 3:29 PM PST
impact duration: 11 hrs. 7 mins

31

Habits we've learned so far at Microsoft





Be Customer Obsessed

Iterate over Pain

Production First Mindset

Team Autonomy + Enterprise Alignment

Shift Left Quality

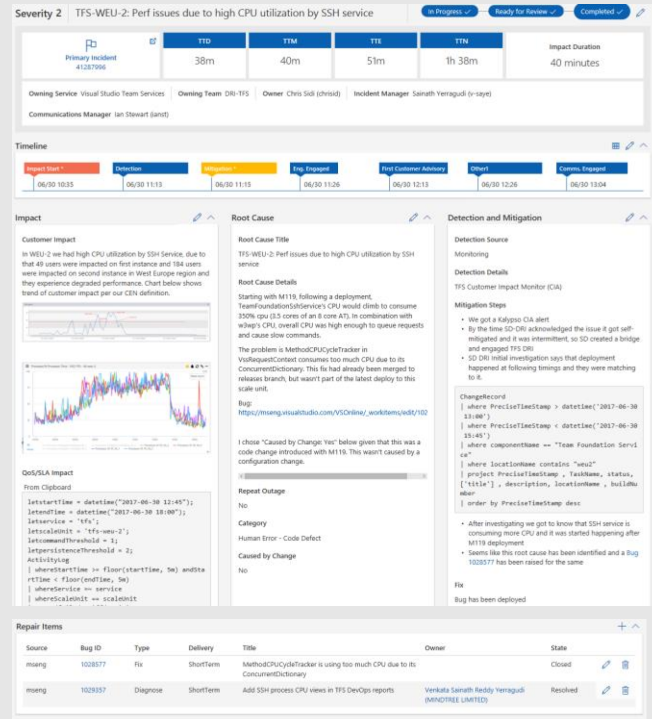
Infrastructure as Flexible Resource

Don't over-think, learn how to fail fast

32

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

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

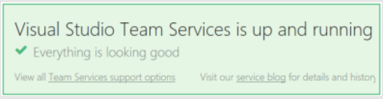
Bug has been deployed

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

33

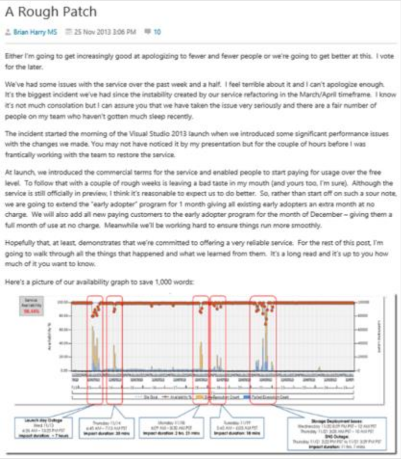
Be Transparent



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.

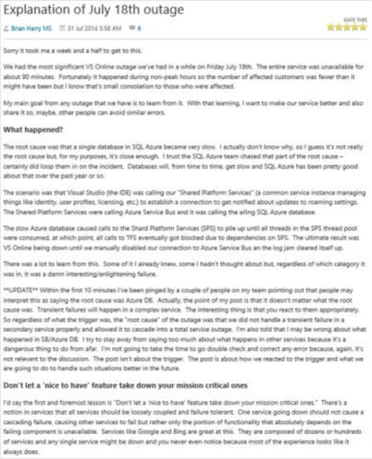
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you 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

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.

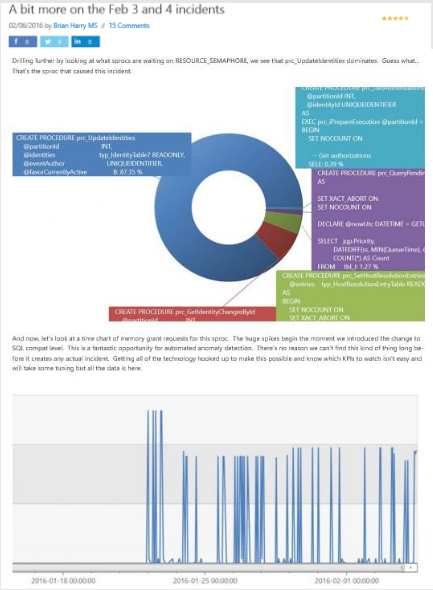
The slow Azure database caused calls to the Shard Platform Services (SPS) to pile up until all threads in the SPS thread pool were consumed, at which point, all calls to TFS eventually got blocked due to dependencies on SPS. The ultimate result was VS Online being down until we manually disabled our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

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.

CREATE PROCEDURE prc_UpdateIdentities
@partitionId INT,
@identities typ_IdentityTable7 READONLY,
@eventAuthor UNIQUEIDENTIFIER,
@favorCurrentlyActive B: 87.35 %
@partitionId INT,
@identityId UNIQUEIDENTIFIER
AS
EXEC prc_iPrepareExecution @partitionId =
BEGIN
SET NOCOUNT ON
-- Get authorizations
SELE: 0.39 %
CREATE PROCEDURE prc_QueryPendin
AS
SET XACT_ABORT ON
SET NOCOUNT ON
DECLARE @nowUtc DATETIME = GETU
SELECT jqp.Priority,
DATEDIFF(ss, MIN(QueueTime),
COUNT(*) AS Count
FROM tbl_J: 1.27 %
CREATE PROCEDURE prc_SetHostResolutionEntries
@entries typ_HostResolutionEntryTable READ
AS
BEGIN
SET NOCOUNT ON
SET XACT_ABORT ON
CREATE PROCEDURE prc_GetIdentityChangesById

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.

34

No such thing as ‘partial automation’

“One time” deployment commands in OneNote, email

```
Set-Options “-p 0”
```

Imagine a dozen more steps like that…

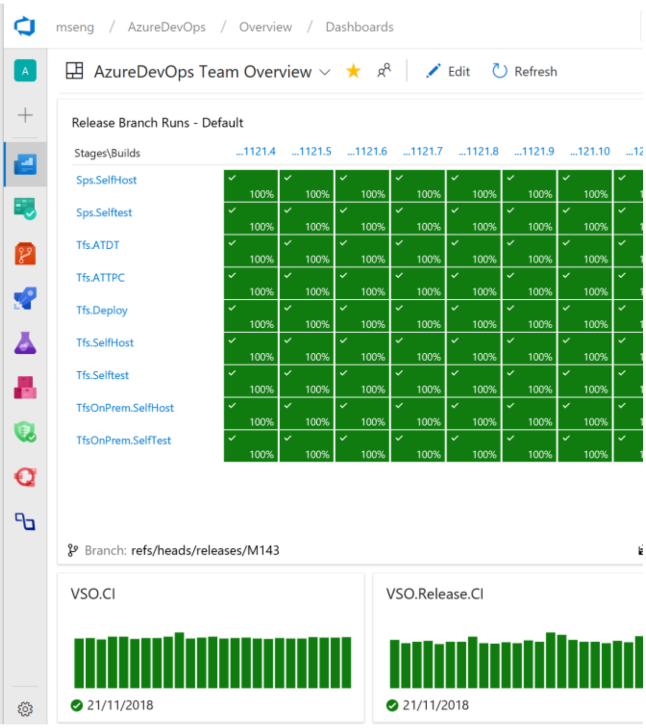
And then…someone misses a step half way through

We once broke pre-production for a day

35

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	...12
Sps.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Sps.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Tfs.ATDT	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Tfs.ATTPC	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Tfs.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
Tfs.Selftest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓

Branch: refs/heads/releases/M143

VSO.CI

✓ 21/11/2018

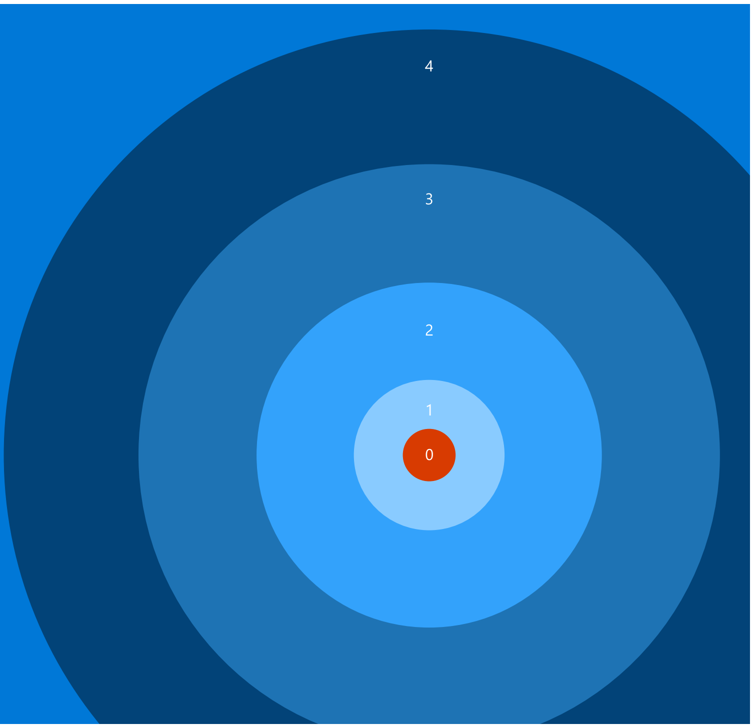
VSO.Release.CI

✓ 21/11/2018

36

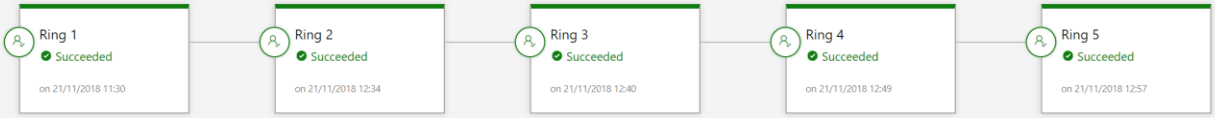
Your aim won't be perfect.

Control the blast radius.



37

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

38

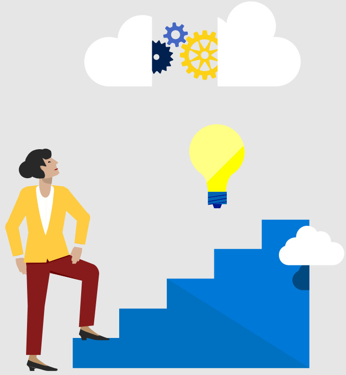
na·ked agil·i·ty
Mar·tin Hin·shel·wood

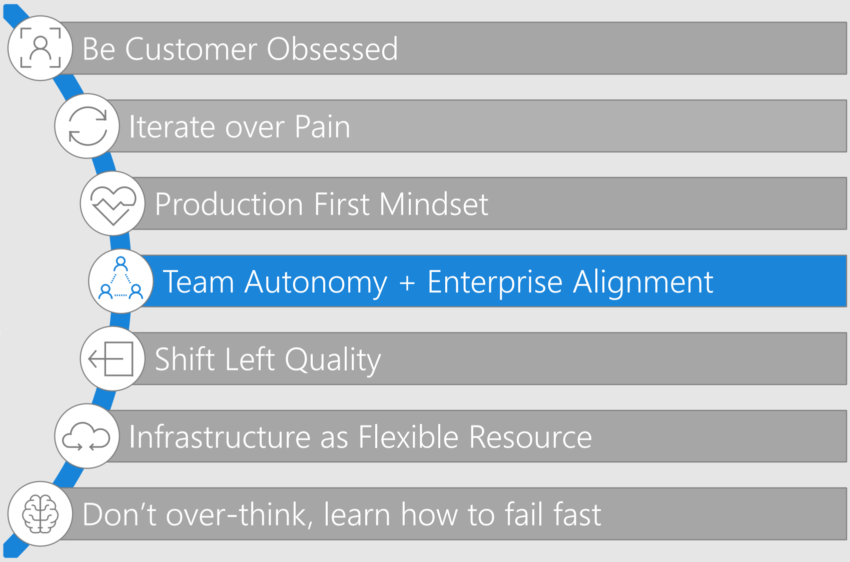
Live Site Culture

- 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

39

Habits we’ve learned so far at Microsoft



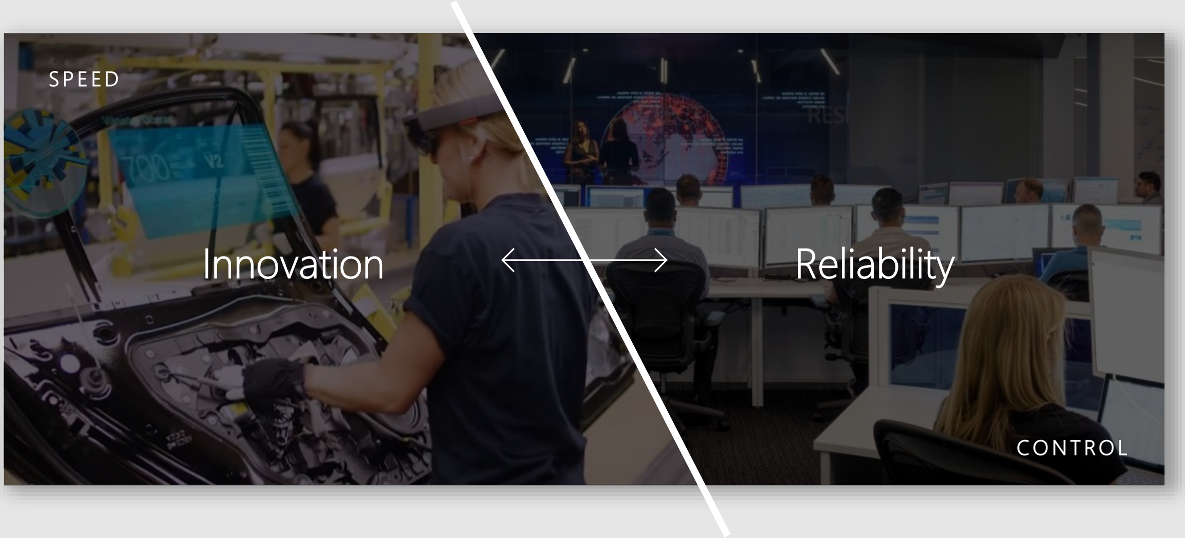


- Be Customer Obsessed
- Iterate over Pain
- Production First Mindset
- Team Autonomy + Enterprise Alignment
- Shift Left Quality
- Infrastructure as Flexible Resource
- Don’t over-think, learn how to fail fast

40

Software delivery paradox

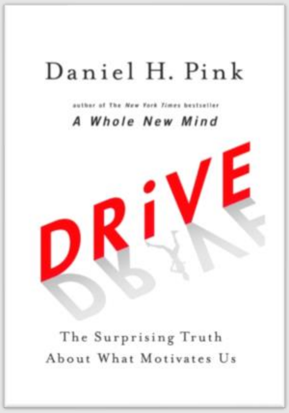
Speed vs. control

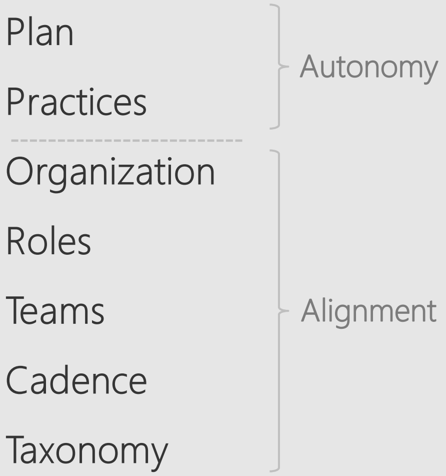


41

Agile at Scale with Aligned Autonomy

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





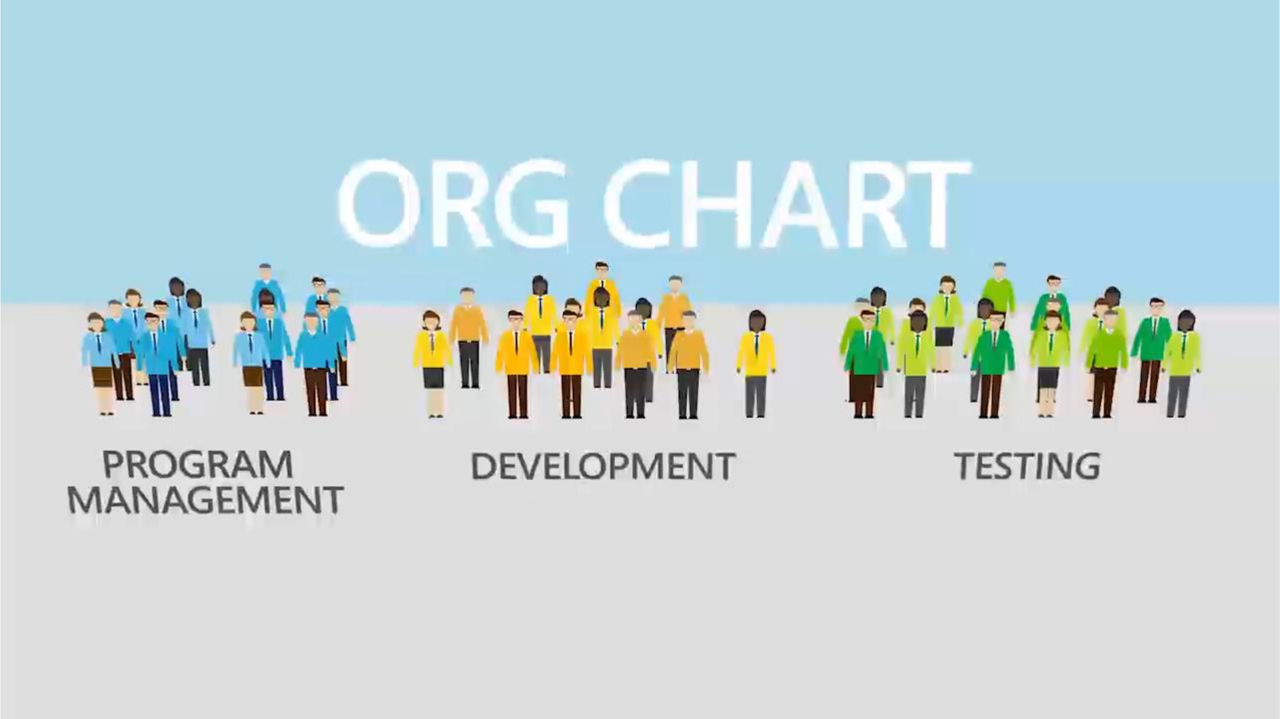
Plan
Practices
Autonomy

Organization
Roles
Teams
Cadence
Taxonomy
Alignment

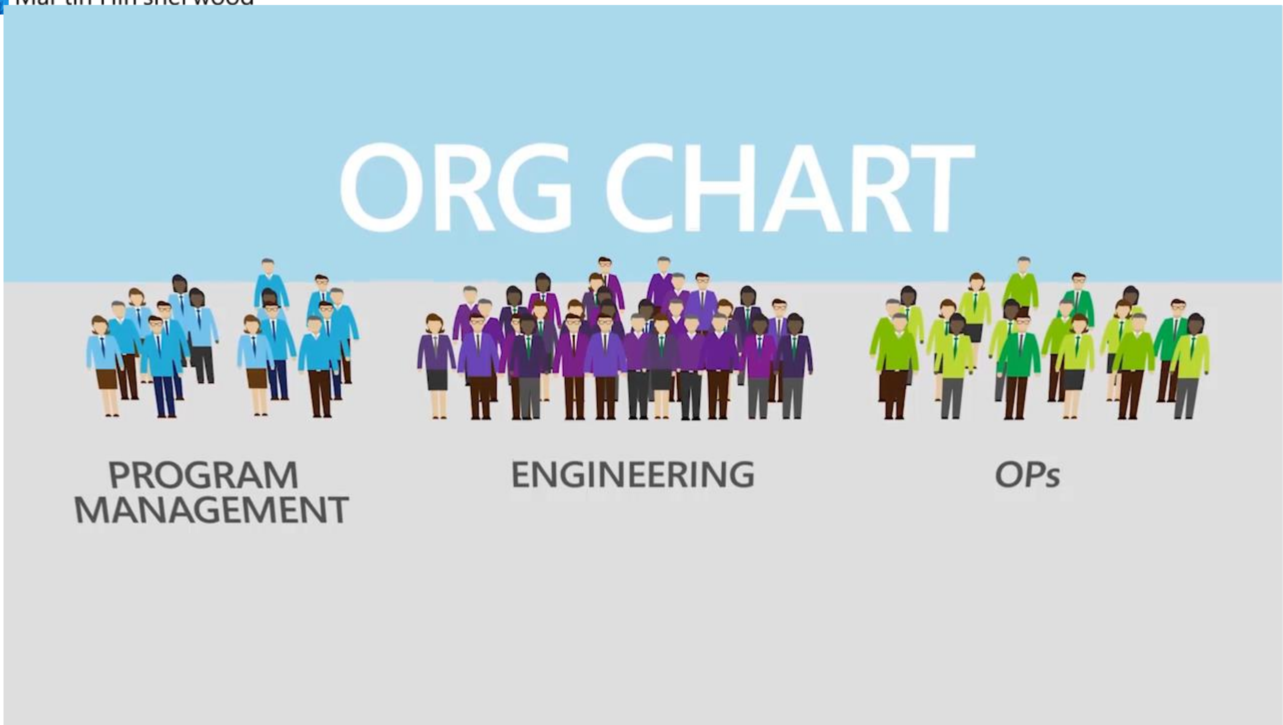
42

Team Structure

43



44



ORG CHART

PROGRAM MANAGEMENT

ENGINEERING

OPs

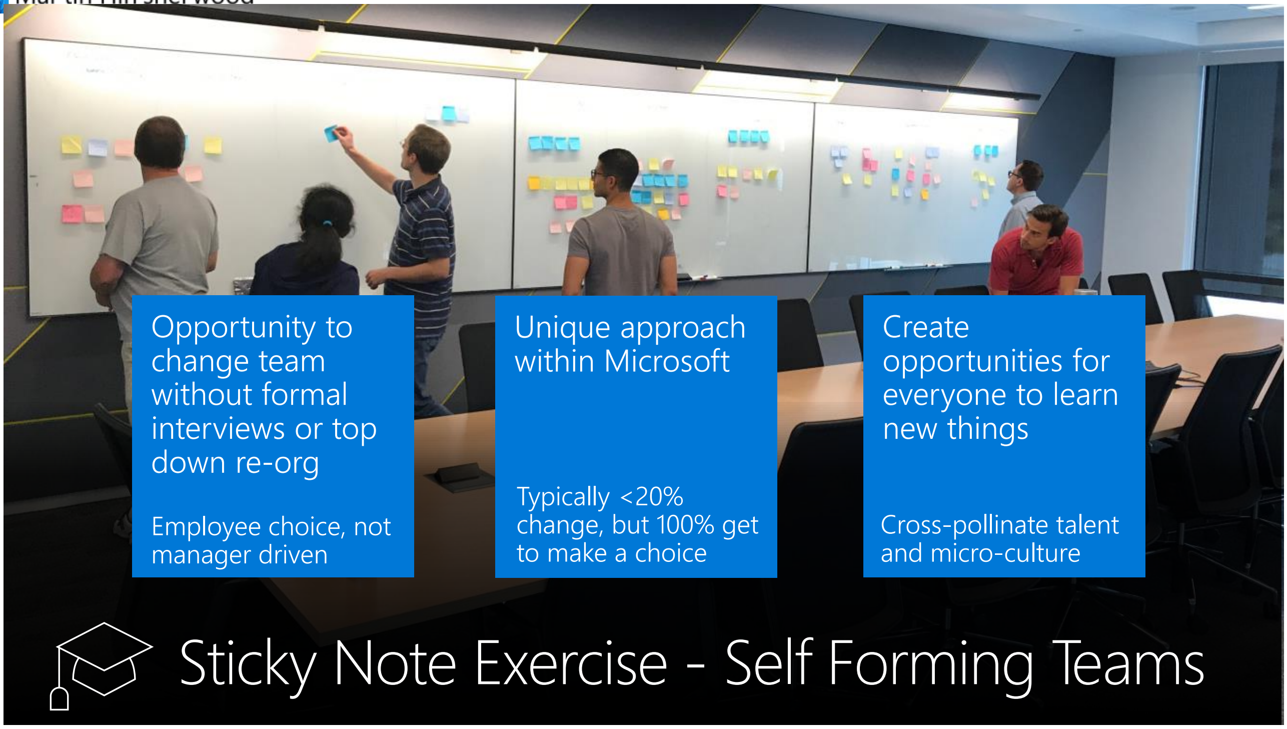
45

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

46

na·ked agil·i·ty
Mar·tin Hin·shel·wood



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

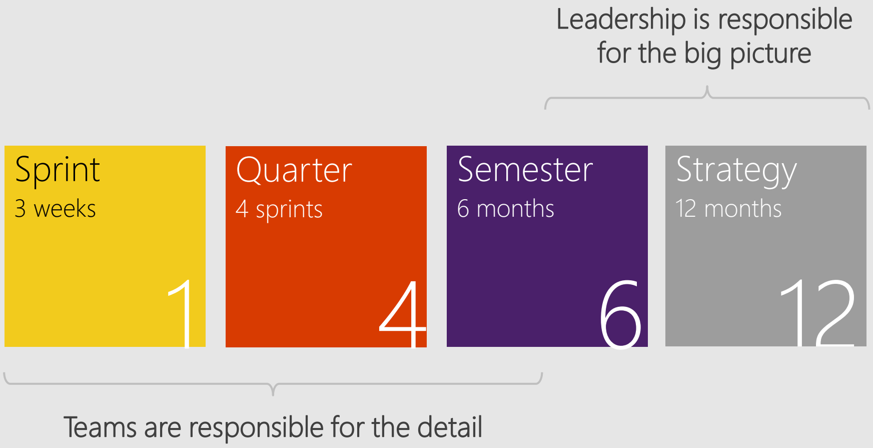
Create opportunities for everyone to learn new things

Cross-pollinate talent and micro-culture

Sticky Note Exercise - Self Forming Teams

47

Planning



Leadership is responsible for the big picture

Sprint
3 weeks
1

Quarter
4 sprints
4

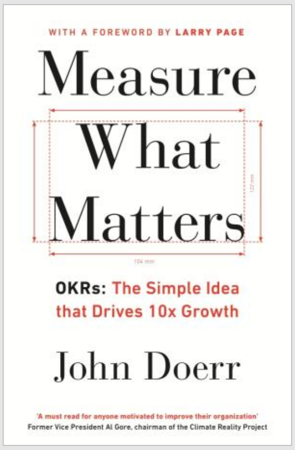
Semester
6 months
6

Strategy
12 months
12

Teams are responsible for the detail

48

Measure Outcomes not Outputs



WITH A FOREWORD BY LARRY PAGE

Measure What Matters

OKRs: The Simple Idea that Drives 10x Growth

John Doerr

'A must read for anyone motivated to improve their organization'
Former Vice President Al Gore, chairman of the Climate Reality Project

OKR: Objective→Key Results

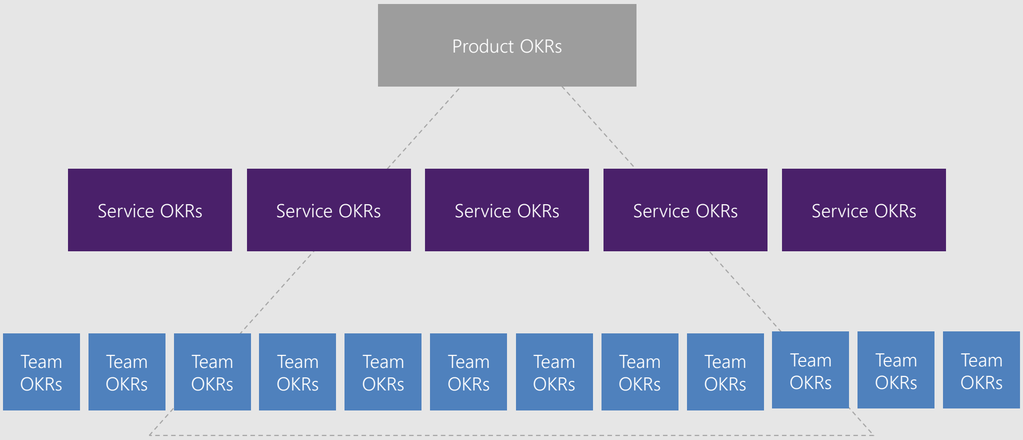
1. Objective: Grow a strong and happy customer base
   - 1.1 Increase external NPS from 21 to 35
   - 1.2 Increase docs SAT from 55 to 65
   - 1.3 New pipeline flow has an Apdex score of 0.9
   - 1.4 Queue time for jobs is 5 seconds or less

KRs are measures for the *quarter*

Encourage ambitious KRs:
70% of the improvement target scores green

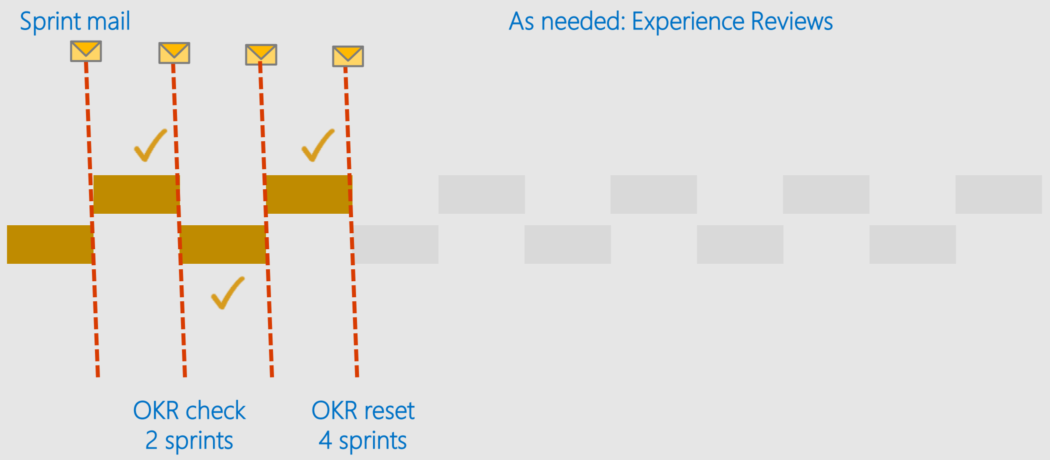
49

Alignment



50

How do you stay in sync?



51

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

52

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

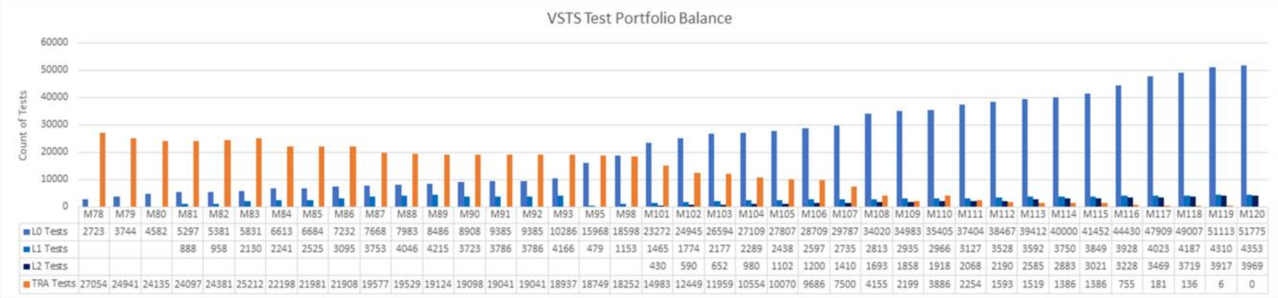
Testing: Shift Left from Integration to Unit

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



VSTS Test Portfolio Balance

	M78	M79	M80	M81	M82	M83	M84	M85	M86	M87	M88	M89	M90	M91	M92	M93	M95	M98	M101	M102	M103
■ L0 Tests	2723	3744	4582	5297	5381	5831	6613	6684	7232	7668	7983	8486	8908	9385	9385	10286	15968	18598	23272	24945	26594
■ L1 Tests				888	958	2130	2241	2525	3095	3753	4046	4215	3723	3786	3786	4166	479	1153	1465	1774	2177
■ L2 Tests																			430	590	652
■ TRA Tests	27054	24941	24135	24097	24381	25212	22198	21981	21908	19577	19529	19124	19098	19041	19041	18937	18749	18252	14983	12449	11959

	M104	M105	M106	M107	M108	M109	M110	M111	M112	M113	M114	M115	M116	M117	M118	M119	M120
■ L0 Tests	27109	27807	28709	29787	34020	34983	35405	37404	38467	39412	40000	41452	44430	47909	49007	51113	51775
■ L1 Tests	2289	2438	2597	2735	2813	2935	2966	3127	3528	3592	3750	3849	3928	4023	4187	4310	4353
■ L2 Tests	980	1102	1200	1410	1693	1858	1918	2068	2190	2585	2883	3021	3228	3469	3719	3917	3969
■ TRA Tests	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

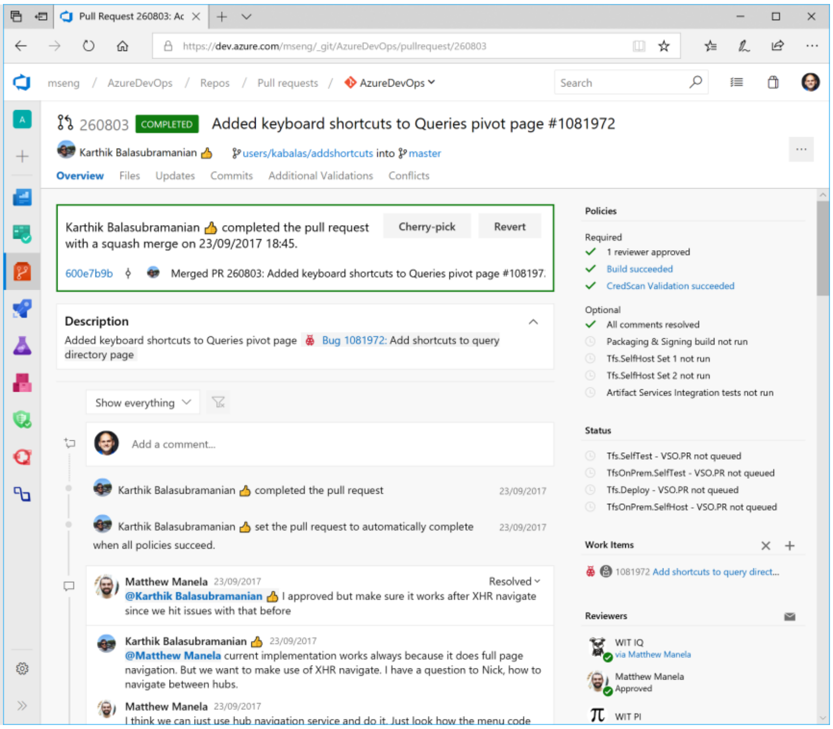
L0+L1 Tests performed before merge

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



260803 COMPLETED Added keyboard shortcuts to Queries pivot page #1081972

Karthik Balasubramanian users/kabalas/addshortcuts into master

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

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

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

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

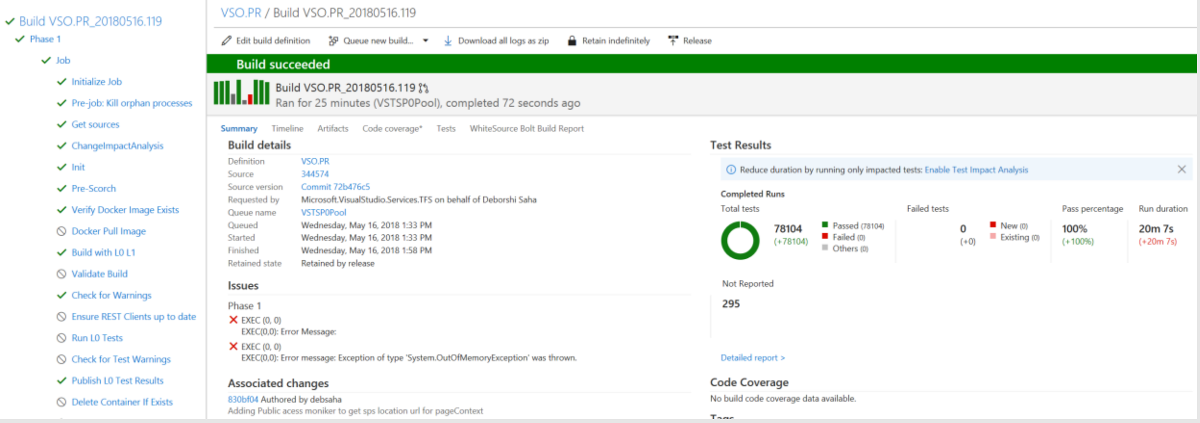
WIT IQ
via Matthew Manela

Matthew Manela
Approved

WIT PI

55

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

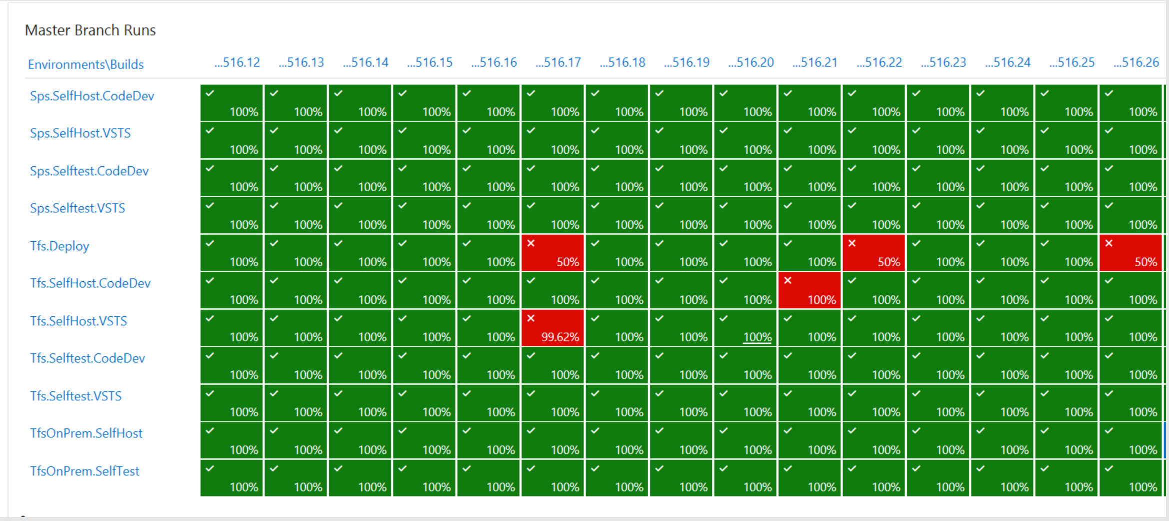
Code Coverage

No build code coverage data available.

Feedback in minutes, before acceptance of PR

56

Green Means Green, Red Means Red



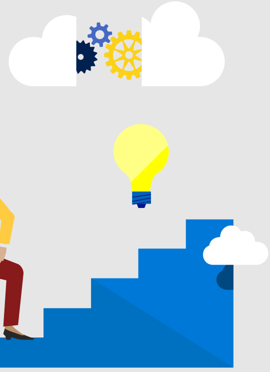
Master Branch Runs

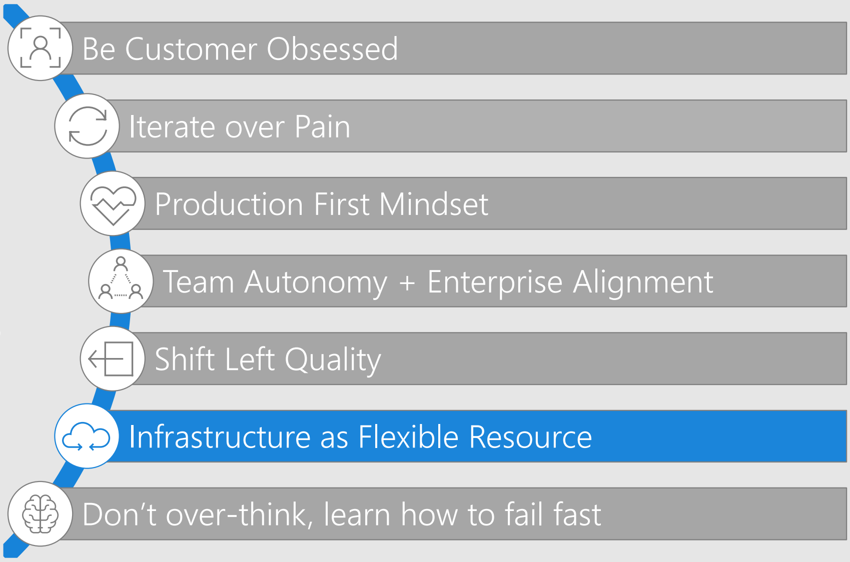
Environments\Builds	...516.12	...516.13	...516.14	...516.15	...516.16	...516.17	...516.18	...516.19	...516.20	...516.21	...516.22	...516.23	...516.24	...516.25	...516.26
Sps.SelfHost.CodeDev	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.SelfHost.VSTS	✓ 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

57

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

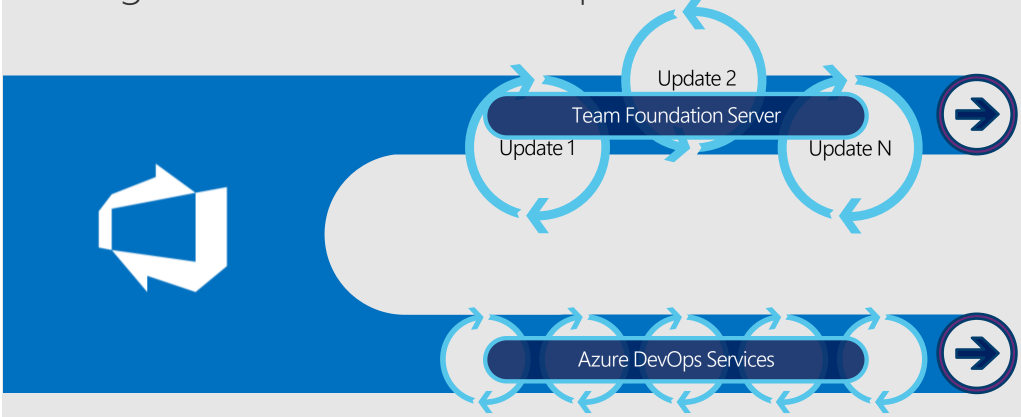
58

Code: Cloud first, then move on-premises

One code base with multiple delivery streams

Shared abstraction layer

Single master branch, multiple release branches



Update 2

Team Foundation Server

Update 1

Update N

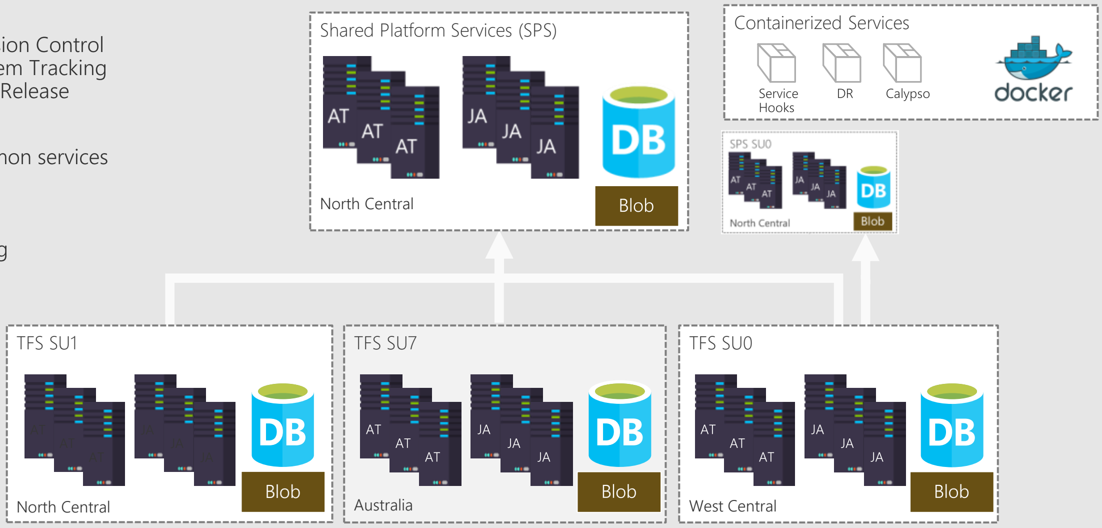
Azure DevOps Services

59

Multiple Data Centers with incremental roll out

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

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



Shared Platform Services (SPS)

AT AT AT JA JA JA DB

North Central

Blob

Containerized Services

Service Hooks

DR

Calypso

docker

SPS SU0

North Central

Blob

TFS SU1

North Central

Blob

TFS SU7

Australia

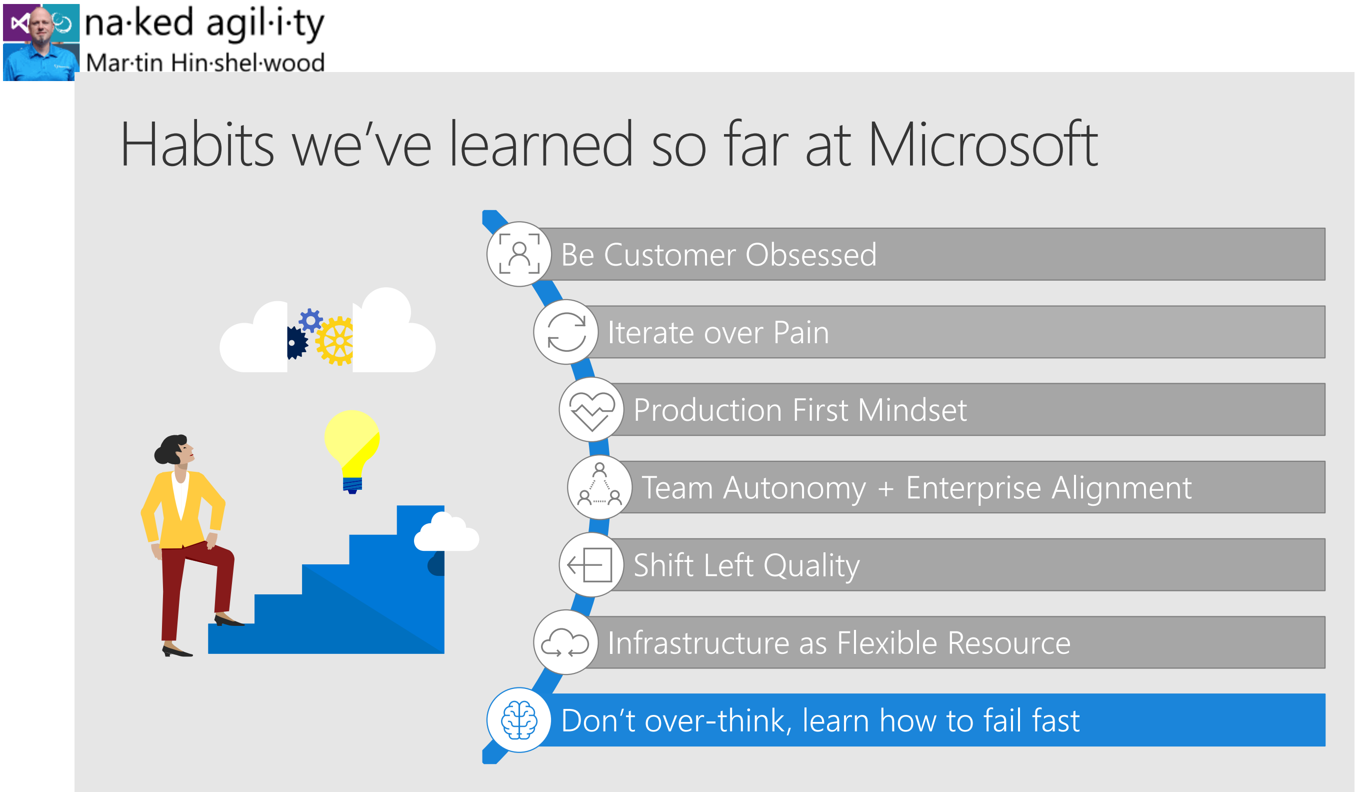
Blob

TFS SU0

West Central

Blob

60

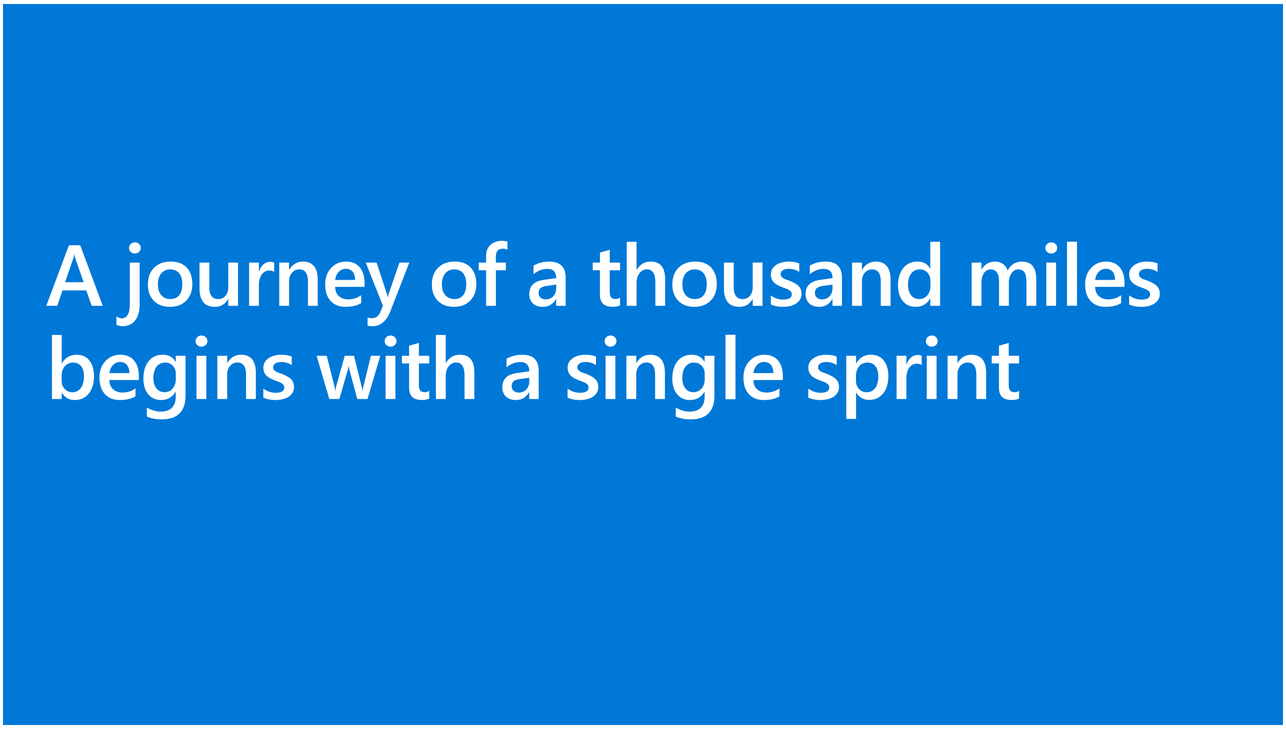


na·ked agil·i·ty
Mar·tin Hin·shel·wood

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

61

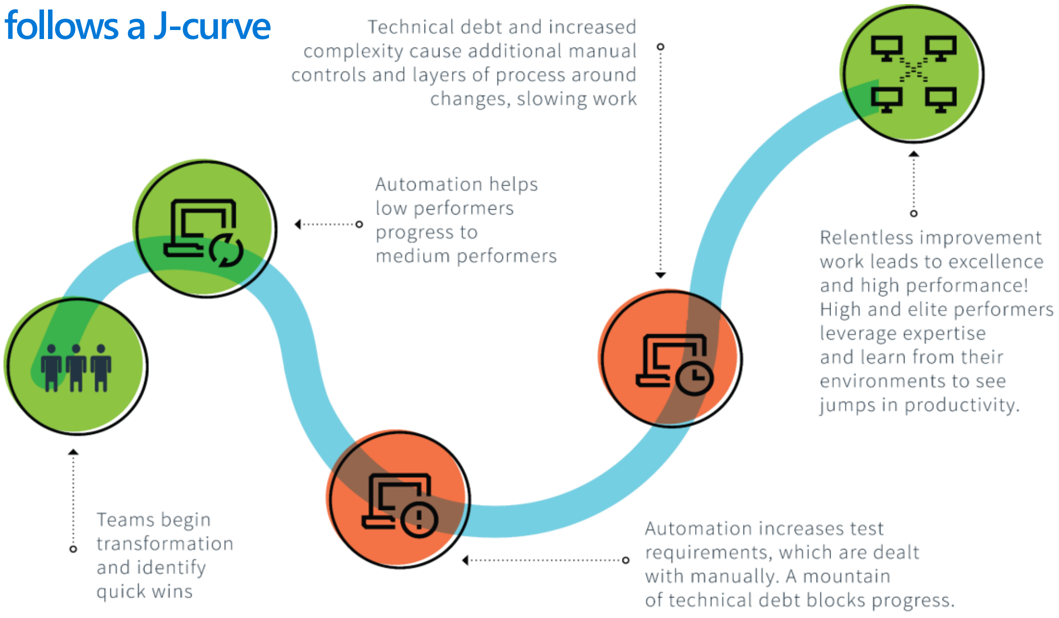


A journey of a thousand miles begins with a single sprint

62

naked agility
Martin Hinshelwood

Progress follows a J-curve



Teams begin transformation and identify quick wins

Automation helps low performers progress to medium performers

Automation increases test requirements, which are dealt with manually. A mountain of technical debt blocks progress.

Technical debt and increased complexity cause additional manual controls and layers of process around changes, slowing work

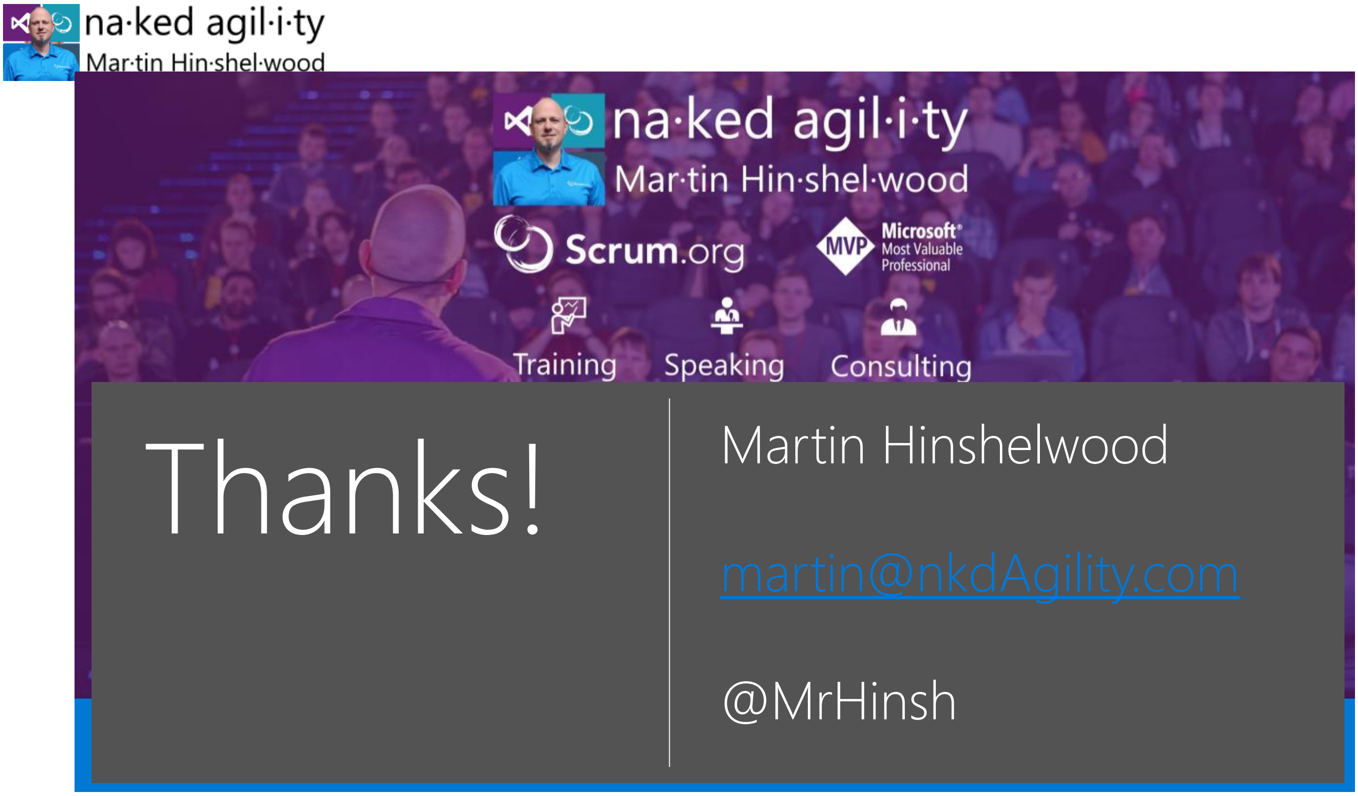
Relentless improvement work leads to excellence and high performance! High and elite performers leverage expertise and learn from their environments to see jumps in productivity.

DORA DEVOPS RESEARCH & ASSESSMENT **Accelerate: State of DevOps 2018:** Strategies for a New Economy

63

DevOps isn't magic

64



Thanks!

Martin Hinshelwood

martin@nkdAgility.com

@MrHinsh

65