

# Mindset Essentials DevOps @ Microsoft Corp.



Willy[-Peter] Schaub  
AJATO Transformations  
ALM | DevOps Ranger

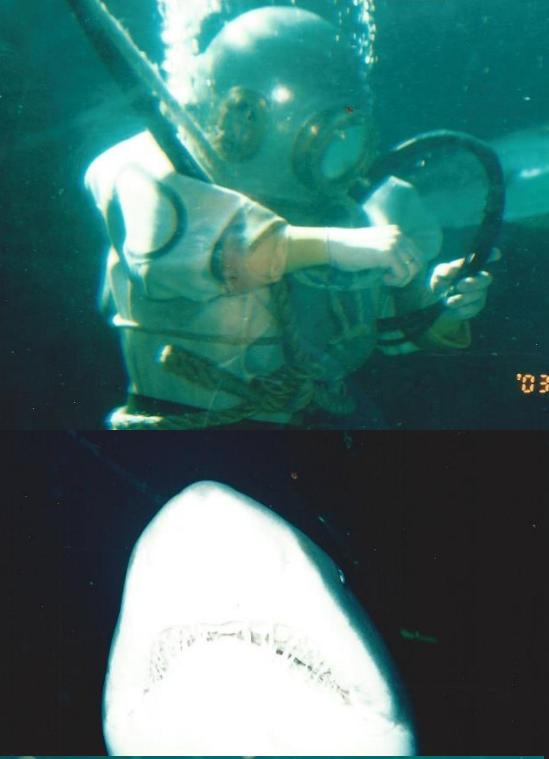




I work  
here

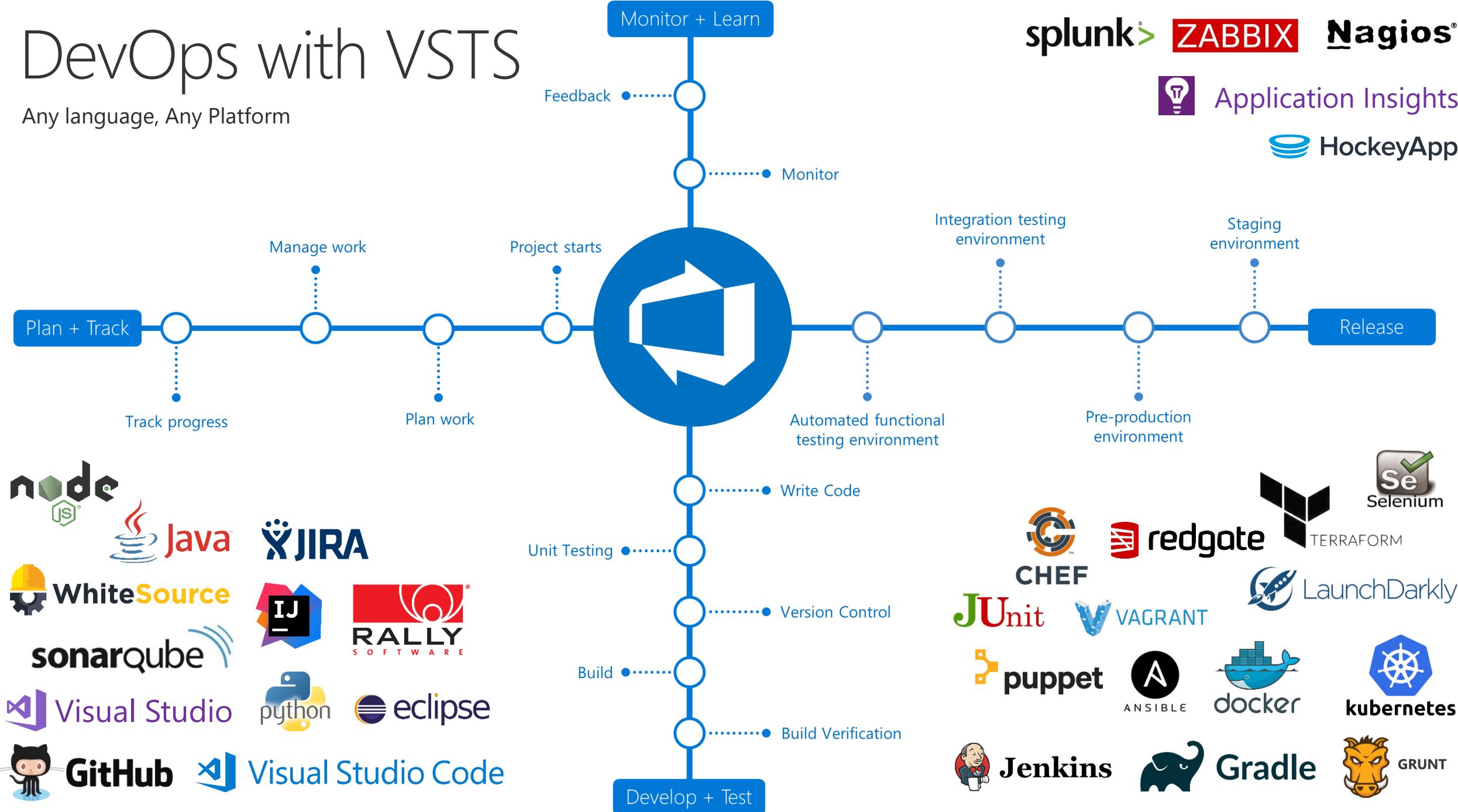
I was born here

I grew up here



# DevOps with VSTS

Any language, Any Platform



# One Engineering System using VSTS

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



# What is DevOps?

DevOps is the union of people, process, and products to enable continuous delivery of value to our customers.



Increase flow of value

Shorten cycle times

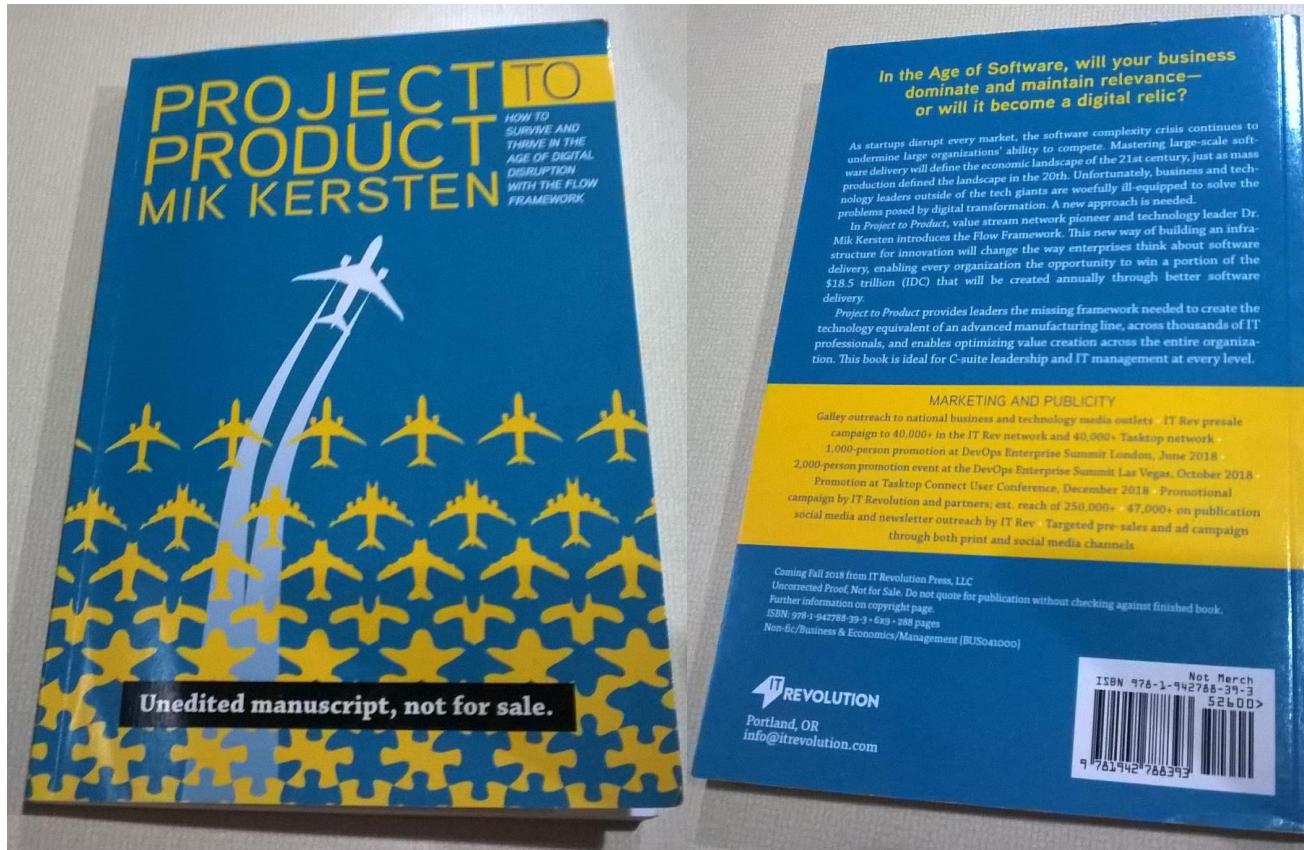
Continuously Improve

# Talking about value ...

Get a copy of this book!

Available for pre-order  
and will be released  
November 20<sup>th</sup>

<http://a.co/d/f7UsTBI>



## Bridging the gap between engineering and business!

# Our Definition of Done

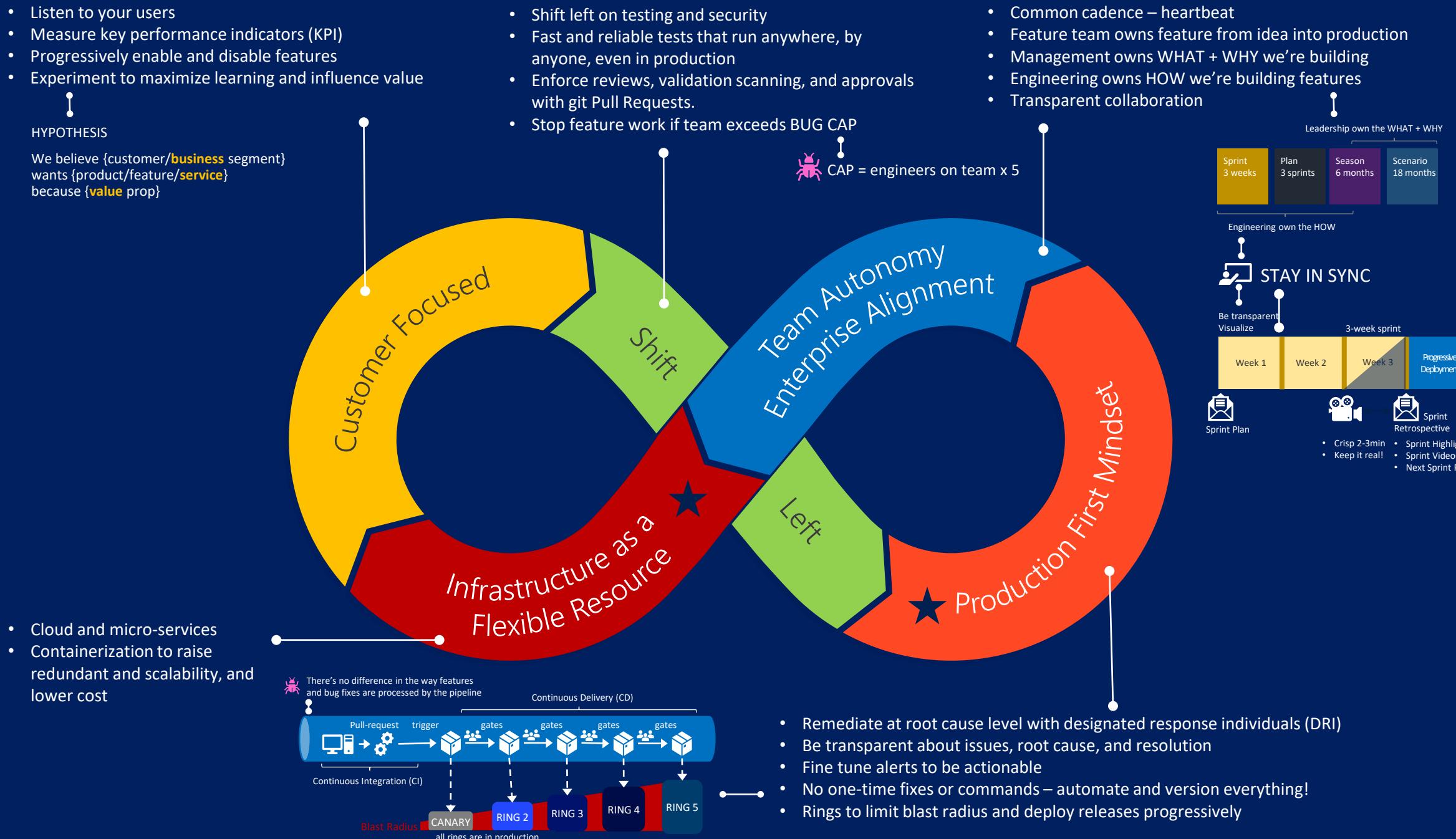
Live in production, collecting telemetry that examines the hypothesis which motivated the deployment.



Our DevOps journey is  
about continuously  
adding value to your  
business and engineering  
system



# DevOps Habits



# Five habits we've learned so far



# Listen to our customers

A screenshot of a Microsoft Edge browser window. The address bar shows [visualstudio.uservoice.com/forums/330519-visual-studio](https://visualstudio.uservoice.com/forums/330519-visual-studio). The page title is "Implement real-time collaboration on the Kanban board". The main content area displays a forum post with 136 votes. The post discusses the need for real-time collaboration on the Kanban board. Below the post, there's a screenshot of the VSTS interface showing a Kanban board with cards and a search bar. A tooltip on the interface says "Report a problem..." and "Provide a suggestion...". The bottom of the page shows recent projects like "FabrikamFiber" and "RadioTFS".

A screenshot of a user voice forum on [visualstudio.uservoice.com](https://visualstudio.uservoice.com). The search bar shows "[vsts]". The results page for the tag "vsts" has 3,797 tagged questions. It features a "Sponsored links for this tag" section with links to create a VSTS account, learn more about VSTS, and keep up with DevOps news. Below this is a brief description of VSTS and a "learn more..." link. The main content is a table showing data for various locations:

Location	Business	Champ	Engaged Users	Monthly Δ	Command Count	Availability	Import?	NPS
Dublin, Ireland	Professional services	aaronha	4468	3188	37,962,631	99.78%	yes	10
Amsterdam, NL	Oil & gas	jeffbe	2144	101	81,118,987	99.82%	yes	9
New York, NY	Financial information / analytics	amitgup	1897	42	65,377,799	99.87%	yes	10
Raleigh, NC	Computer assisted legal research	roferg	1794	-8	23,119,242	99.97%	no	8
London, England	Professional services	midenn	1578	720	19,035,301	99.88%	yes	10
Munich, Germany	Reinsurance	ehofman	1204	27	37,827,300	99.79%	yes	9
Seattle, WA	Freight forwarding service	midenn	1187	6	64,852,531	99.92%	no	9
New York, NY	Professional services	daazose	1165	244	46,686,753	99.84%	no	9
Norway, NL, Houston	Oilfield services	samgu	1026	436	98,430,316	99.95%	no	7.5
Bentonville, AR	Retail	buckh	935	118	16,410,929	99.75%	yes	10
Charleston, SC	Software publishing	chrispat	927	96	28,950,208	99.88%	yes	9
Salt Lake City, UT	IT Support Services	gauravsi	828	82	42,601,059	99.90%	yes	
Peoria, IL	Heavy equipment manufacturing	abarr	746	117	6,893,030	99.91%	no	8
Palo Alto, CA	Computer hardware & software	marirod	742	-36	7,913,156	99.87%	no	8
London, England	Retail clothing	angelp	706	80	76,135,638	99.44%	no	
Minnetonka, MN	Agriculture	taylaf	575	-1	12,056,989	99.61%	yes	10
Seattle, WA	IT Consulting	aaronha	574	119	5,621,066	99.95%	no	
Scottsdale, AZ	Project administration software	tarekm	556	-7	44,387,813	99.86%	yes	
Liberty Lake, WA	Energy and water resource mgmt	buckh	556	49	14,421,983	99.85%	no	8

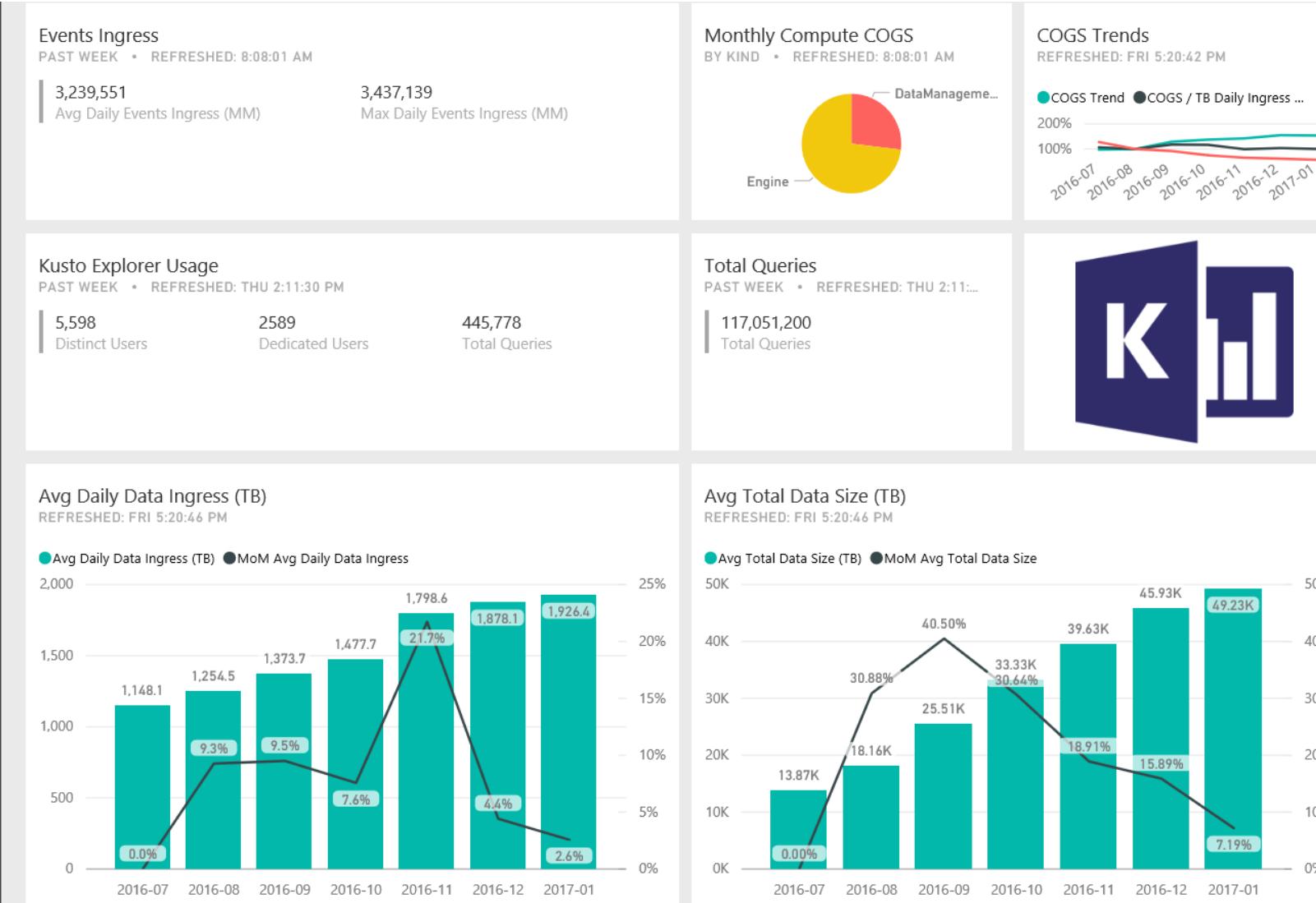
Customer Focused

# Collect data broadly (but carefully)

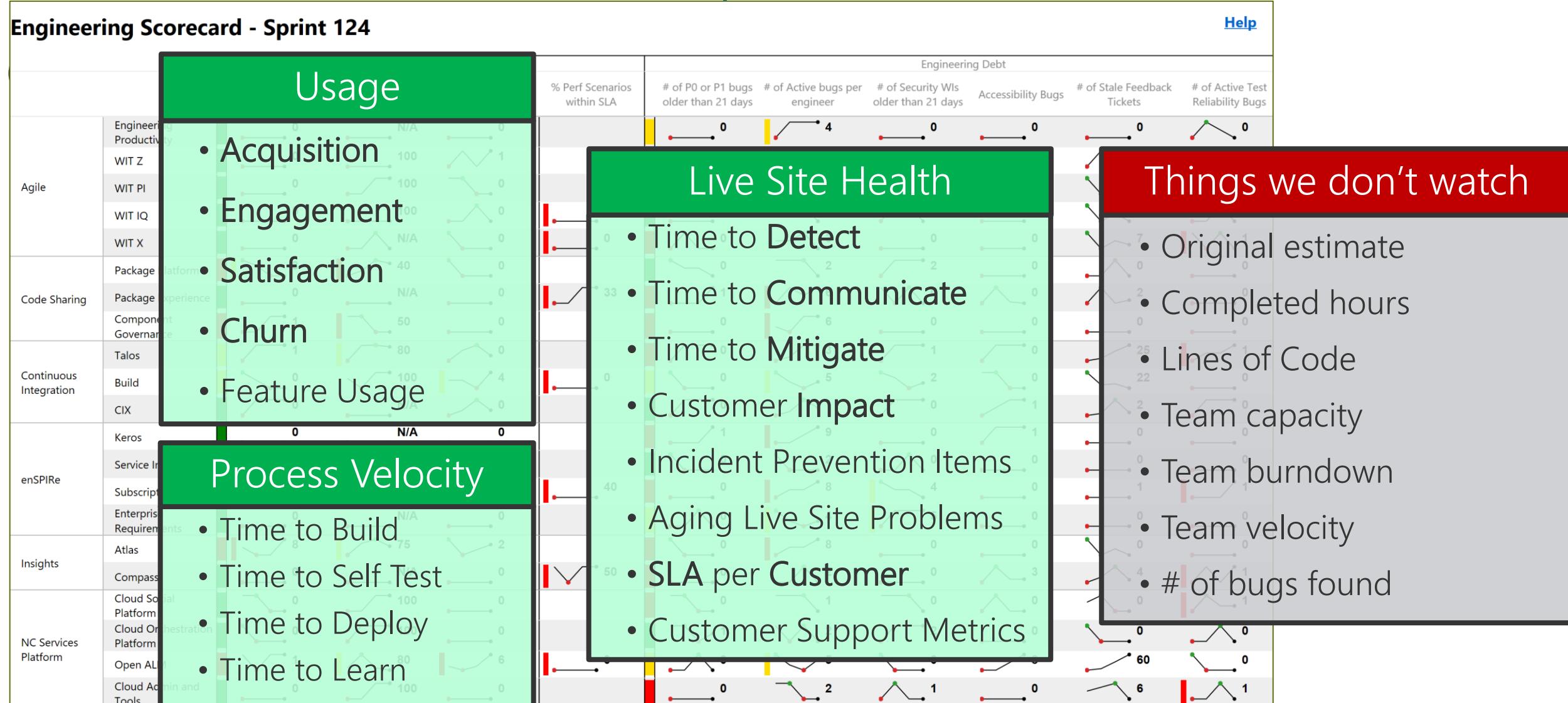
## Application Insights Analytics (Project Kusto) for

- 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			
1	MegaByte	= 1,000 KiloByte	1,000,000
1	GigaByte	= 1,000 MegaByte	1,000,000,000
1	TerraByte	= 1,000 GigaByte	1,000,000,000,000
1	PetaByte	= 1,000 TerraByte	1,000,000,000,000,000
			15



# But measure what's important (KPI's)

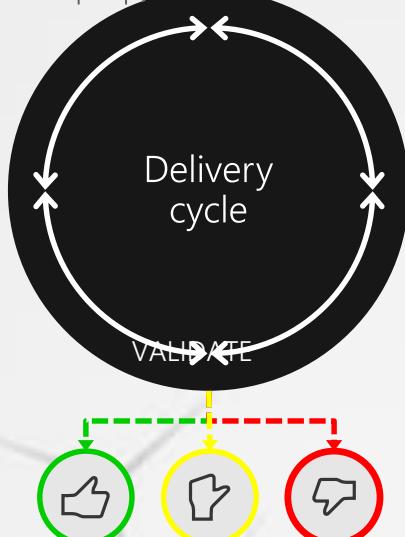


# Maximize learning and influence value

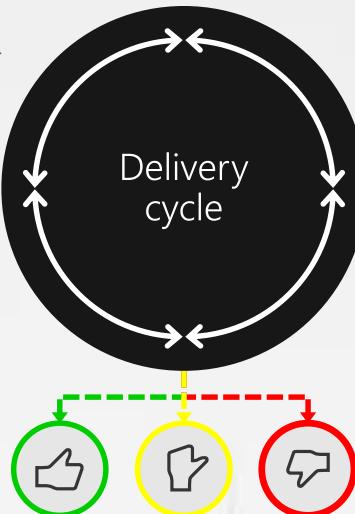
Validated learning ↑

## HYPOTHESIS

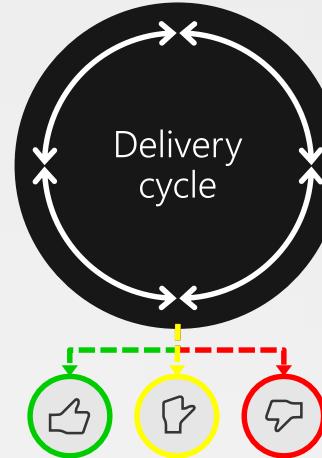
We believe {customer/business segment} wants {product/feature/service} because {value prop}



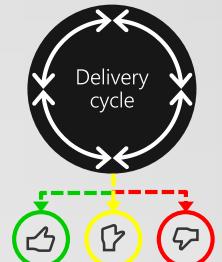
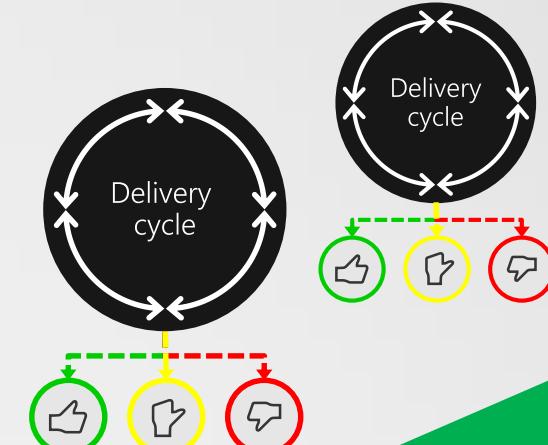
PROVEN INCONCLUSIVE DISPROVEN



Deployment frequency →



Value of improvement



# Feature Flags – fine tune user experience

All code is deployed - feature flags control exposure

Reduces integration debt

Flags provide runtime control down to individual user

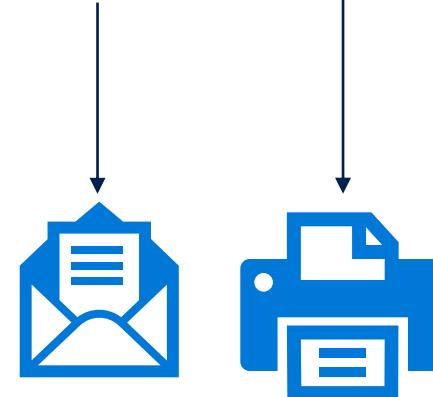
Users can be added or removed with no redeployment

Progressive experimentation & refinement

Enables dark launch



if ( flag )  
else



# EPIPHANY #1

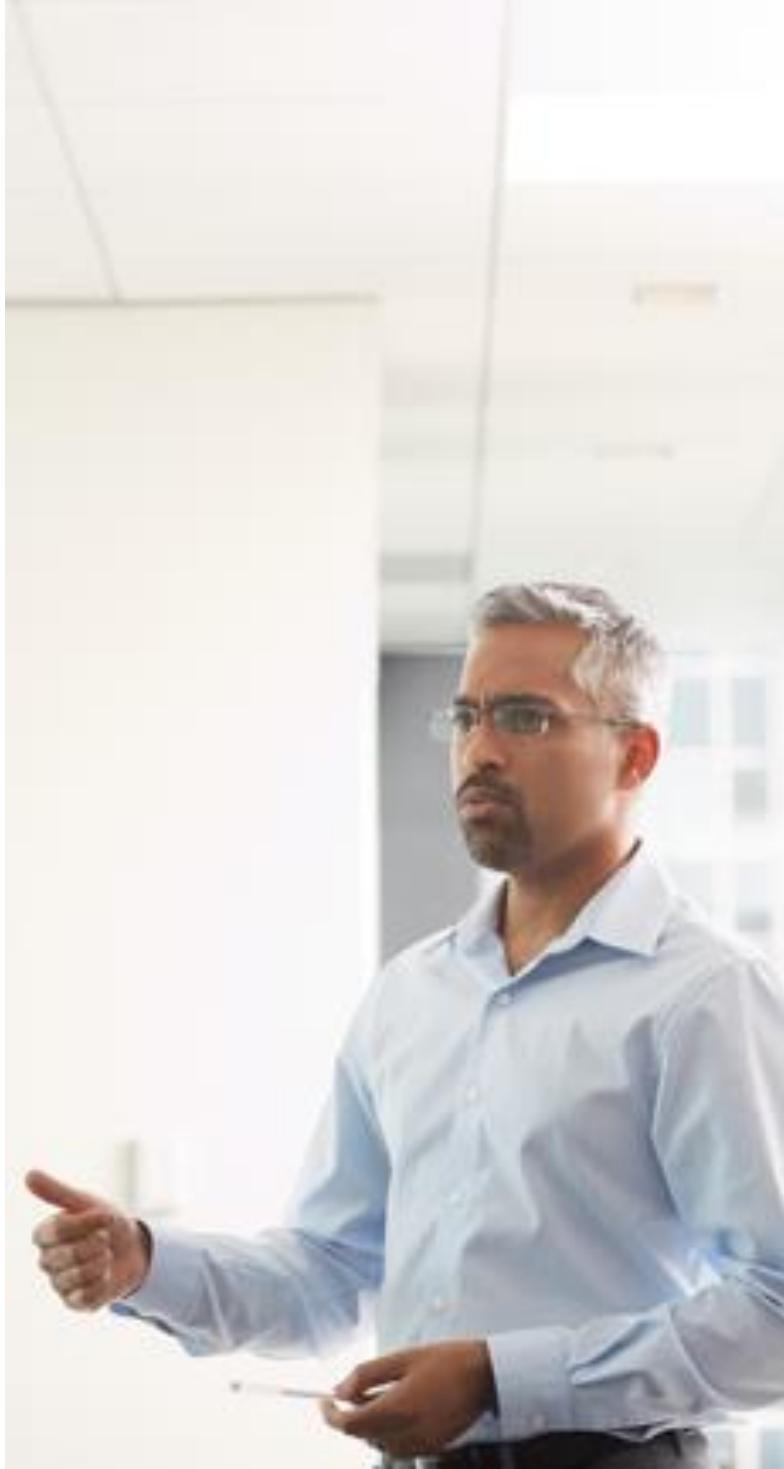
Feature flags are extremely valuable and powerful ... if used correctly and technical debt is managed.

Feature flags are not intended to hide code in production that is **not** production ready.



Image from elyob, flicker

# Five habits we've learned so far



# Live Site Incidents

On detection, LSI conference bridge created

DRI's brought in to call

Gather data for root cause analysis

Mitigate!

5-15min

Every action recorded

Rotate people during long running LSI's

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	TTD	TTM	TTE	TTN	Impact Duration
41287996	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 \* 06/30 10:35 Detection 06/30 11:13 Mitigation \* 06/30 11:15 Eng. Engaged 06/30 11:26 First Customer Advisory 06/30 12:13 Other! 06/30 12:26 Comms. Engaged 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, TeamFoundationShService'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/1028577](https://mseng.visualstudio.com/VSONline/_workitems/edit/1028577)

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.

**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

**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

# Be Transparent

## 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.

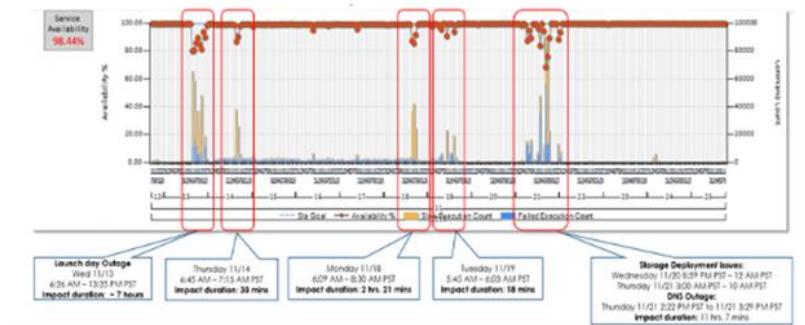
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you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

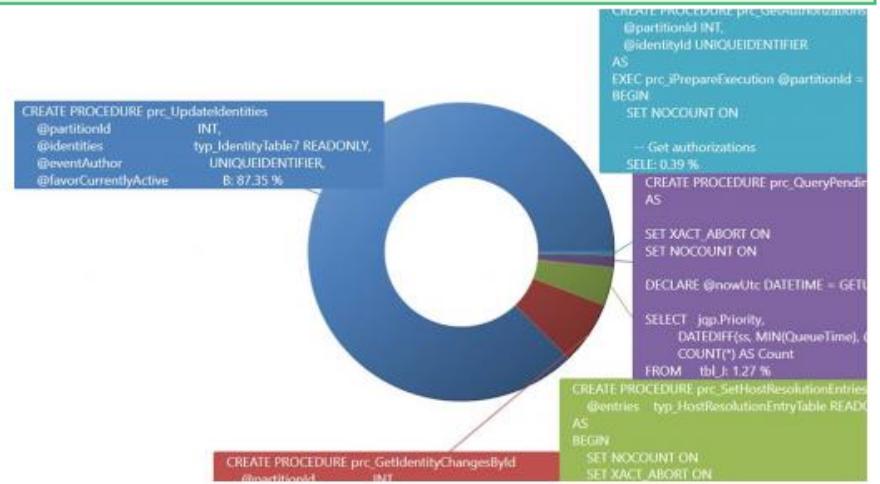
I'd say the first and foremost lesson is "Don't let a 'nice to have' feature take down your mission critical ones." There's a notion in services that all services should be loosely coupled and failure tolerant. One service going down should not cause cascading failure, causing other services to fail 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.

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 history



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.



# Alerting is the key to fast detection

## Before

- Redundant alerts for same the issue
- Needed to set **right thresholds** and tune often
- **Stateless** alerts contributed to further noise

## After

- Every alert must be **actionable** and represent a real issue with the system.
- Alerts should create a **sense of urgency** – false alerts dilutes that
- Use alerts to auto-dial the DRLs.



# Automate completely

No more “one time” commands run manually

Every command goes in PowerShell scripts that are checked in

Deployment to pre-prod & canary is the same as deployment to production every time

All orchestrated with VSTS Release Management

The screenshot shows the VSOOnline dashboard with the following sections:

- Release Branch Runs - Trial Phase in S107**: A grid showing deployment status across environments (Ring 0, Ring 1, PPE Binary, Prod Binary) and various hosts (Sps.SelfTest, Sps.SelfHost, Tfs.SelfHost Set 1, Tfs.SelfHost Set 2, Tfs.SelfTest, Tfs.Deploy, TfsOnPrem.SelfHost). Most cells are green (100% success), except for a few red and yellow ones.
- TFS - Prod Update**: A summary of deployment status for four rings (Ring 0, Ring 1, Ring 2, Ring 3). Each ring has a green button labeled "TFS - Prod..." with a checkmark. A "+1" button is shown to the right.
- VSO.Release.CI**: A chart showing green bars representing successful builds over time, with a note "11/1/2016".
- VSO.Package.RealSign**: A chart showing green and red bars representing successful and failed package real signs over time, with a note "11/1/2016".

At the top, there are navigation links: Home, Code, Work, Build & Release, Test, Wiki, Compliance \*, and a gear icon. A message at the top says: "We've made big improvements to the navigation experience in Team Services. [Take the tour](#) or view the [release notes](#) for more details."

# Tracking Deployments to Production

VSOOnline / VSOOnline Te... ▾ Dashboards Code Work Build and Release Test Wiki Governance\* Analytics Engineering \*\*\* ⚙

Builds Releases Releases\* Library Task Groups Deployment Groups Packages WhiteSource Bolt

New release progress views: Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

⟳ + Search release definitions... 🔎

Release Definitions

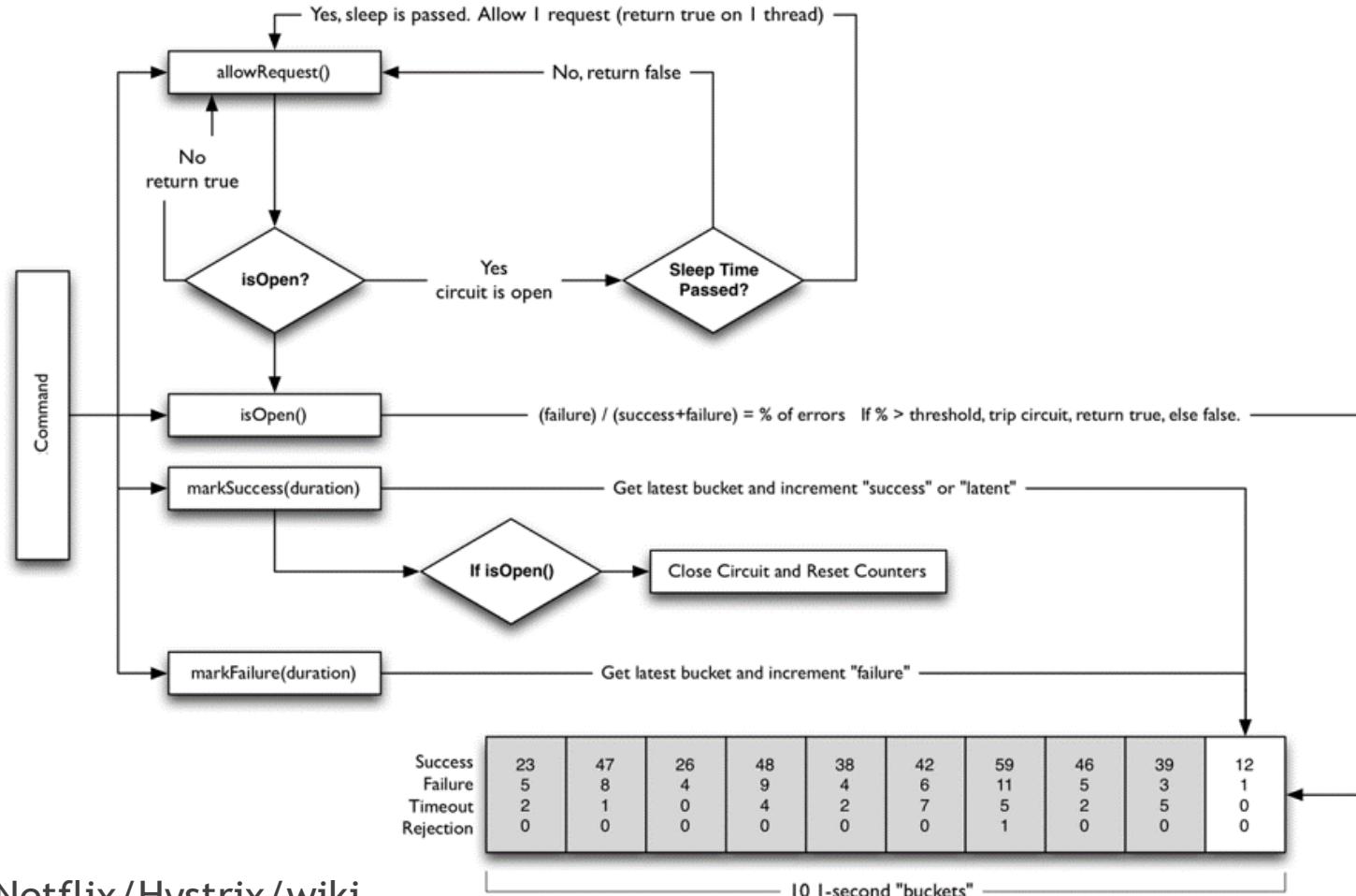
All release definitions

Lock	⚡ Title	Environments	Build	Branch	Created ↓	Created By	Description
🔒	TFS - Prod Config Change 6264	...	VSO.Release.CI_M13 releases/M132	an hour ago	Aldo Mendoza Sa...	NuGet build task hotfix	
🔒	TFS - Prod Config Change 6263	✓	VSO.Release.CI_M13 releases/M133	an hour ago	Aldo Mendoza Sa...	Npm build task hotfix	
🔒	TFS - Prod Config Change 6262		VSO.Release.CI_M13 releases/M132	an hour ago	Jiange Sun	Enable concurrent clone throttli...	
🔒	TFS - Prod Config Change 6261		VSO.Release.CI_M13 releases/M133	an hour ago	Keith Robertson ...		
🔒	TFS - Prod Config Change 6260	✗	VSO.Release.CI_M13 releases/M133	4 hours ago	Ryan Simmen	Block Edwin in TFS PF for DRI Tr...	
🔒	Draft-6586		VSO.Release.CI_M13 releases/M133	6 hours ago	Edwin Sirkko		
🔒	TFS - Prod Config Change 6259		VSO.Release.CI_M13 releases/M133	6 hours ago	Edwin Sirkko	set RU limits to 4x for almengin...	
🔒	TFS - Prod Config Change 6258		VSO.Release.CI_M13 releases/M133	9 hours ago	Adam Barr (VSNC)	Enable fault agents in PF ATs	
🔒	TFS - Prod Config Change 6257	✓	VSO.Release.CI_M13 releases/M133	13 hours ago	Roshan Kumar		
🔒	TFS - Prod Config Change 6256	...	VSO.Release.CI_M13 releases/M133	17 hours ago	Harsh Jain		
🔒	TFS - Prod Config Change 6255	✓	VSO.Release.CI_M13 releases/M133	18 hours ago	Harsh Jain		
🔒	TFS - Prod Config Change 6254		VSO.Release.CI_M13 releases/M132	Tuesday	Aaron Zhou		
🔒	TFS - Prod Config Change 6253	✓	VSO.Release.CI_M13 releases/M133	Tuesday	Sadjad Bahmanp...	Enabling VisualStudio.Services.l...	
🔒	TFS - Prod Config Change 6252	✓	VSO.Release.CI_M13 releases/M133	Tuesday	Pavel Iakovenko		
🔒	TFS - Prod Config Change 6251	✗	VSO.Release.CI_M13 releases/M133	Tuesday	Hao Jiang (VSNC)	upgrade one database in tfs-wu...	
🔒	TFS - Prod Config Change 6250	✓	VSO.Release.CI_M13 releases/M132	Tuesday	Lova Kumar Katik...	Get BDs count	
🔒	TFS - Prod Config Change 6249	✓	VSO.Release.CI_M13 releases/M132	Tuesday	Lova Kumar Katik...	Get BD folders count	
🔒	TFS - Prod Config Change 6248	✓	VSO.Release.CI_M13 releases/M132	Monday	David Wilson (VS...	Re-enable new web platform fo...	
🔒	TFS - Prod Config Change 6247	✓	VSO.Release.CI_M13 releases/M132	Monday	Sadjad Bahmanp...	Enabling VisualStudio.Services.l...	
🔒	TFS - Prod Config Change 6246	✓	VSO.Release.CI_M13 releases/M132	Monday	Sean Lumley	Disable new platform	

1 2 3 4 5 6

# Circuit Breakers

If we can't prevent failure – can we limit the impact?



# Security Mindset

Assume Breach - Use War Games to learn attacks and practice response



VS.



- ▶ Double blind test
- ▶ Full disclosure at or near end

- ▶ Share tactics & lessons learned
- ▶ Continued evolution

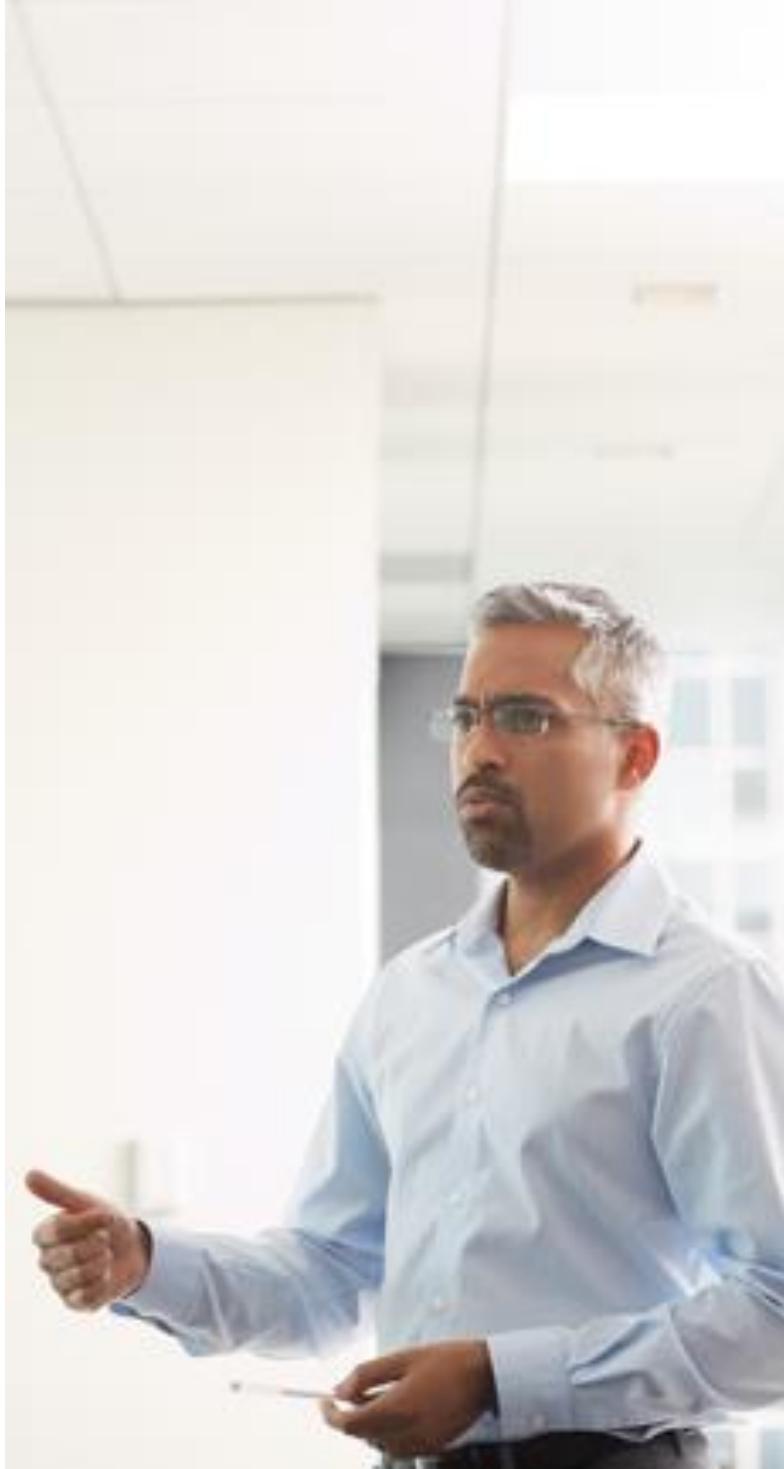
## EPIPHANY #2

The 2AM wake up call is the best motivation for a production ready mindset.

Pull any engineer into an early morning live site incident a few times and the quality bar is magically raised.

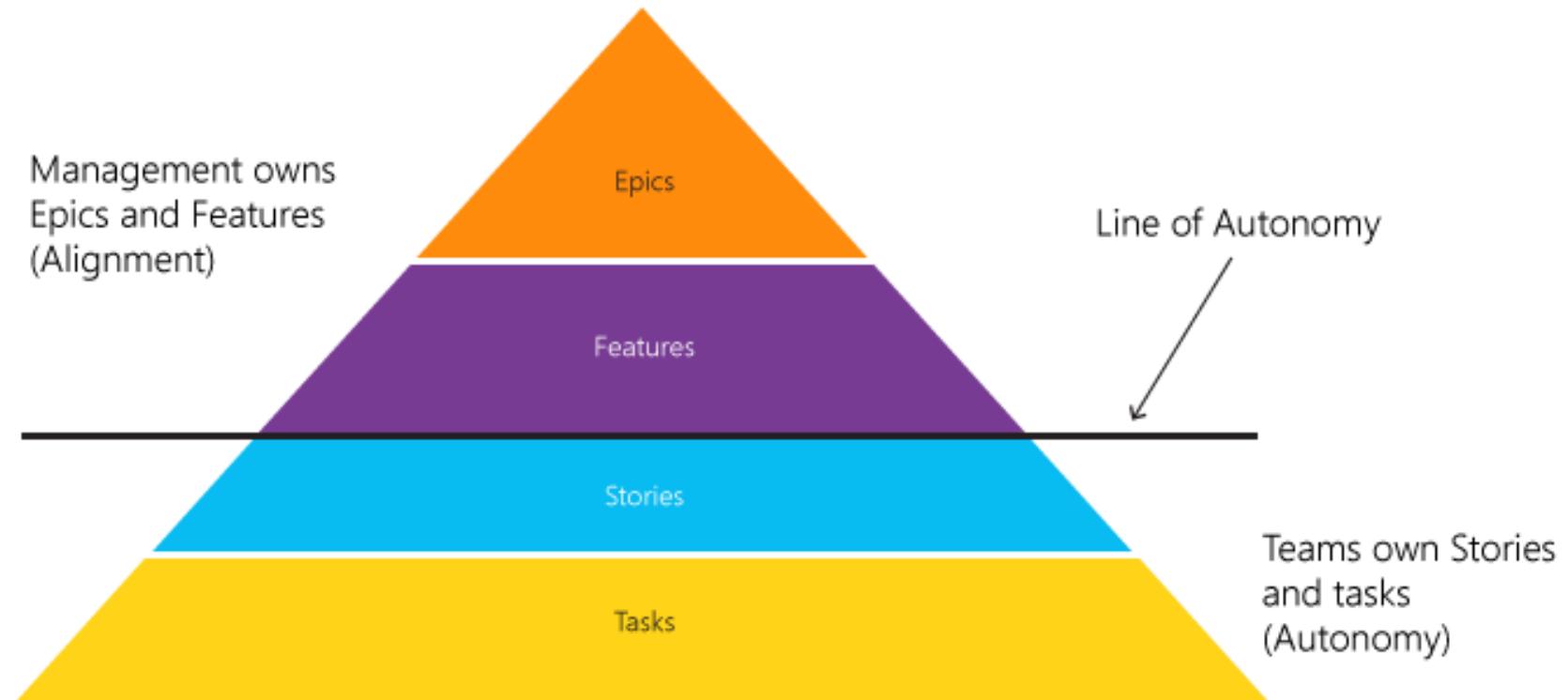
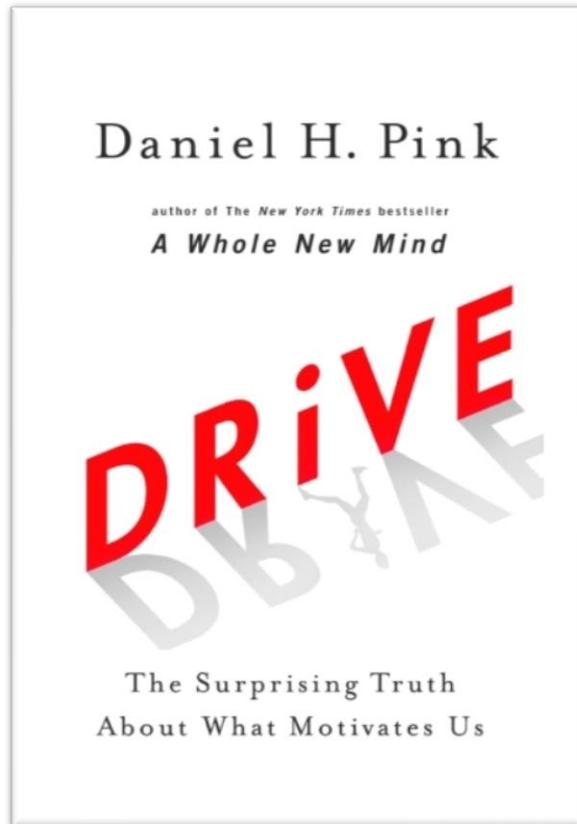


# Five habits we've learned so far



# Agile at Scale with Aligned Autonomy

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

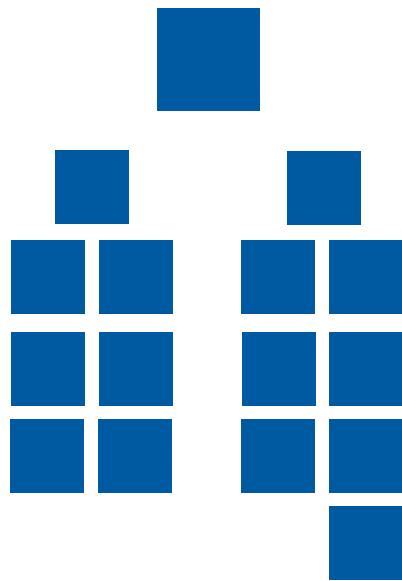


# THE BUG WIT

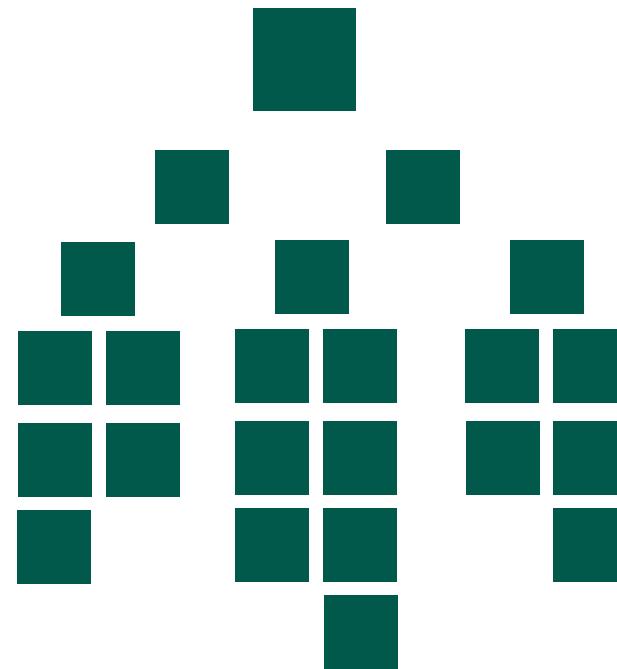
# Roles

The OLD way

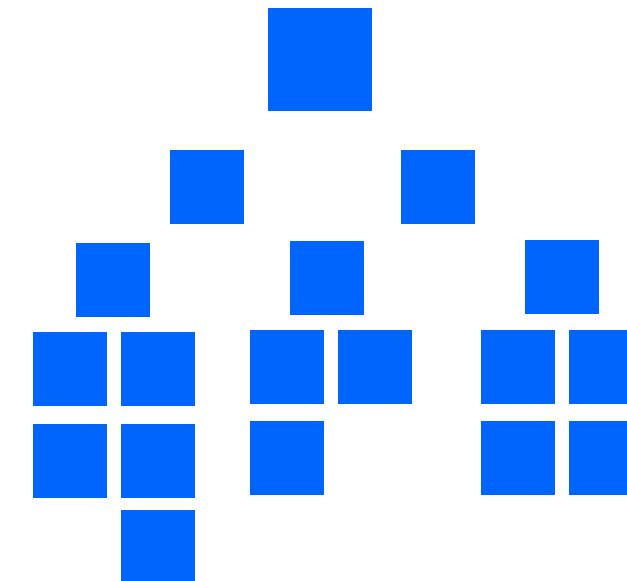
Program Management



Dev

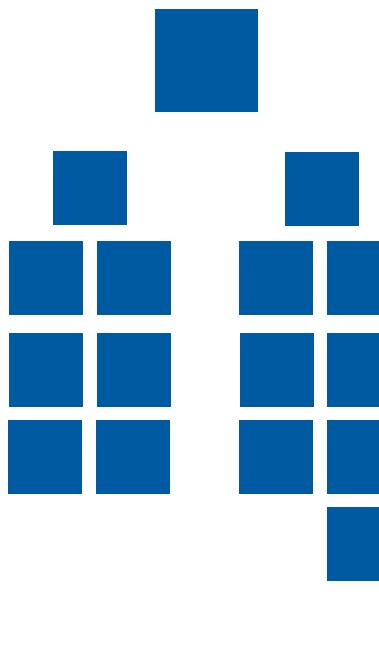


Test

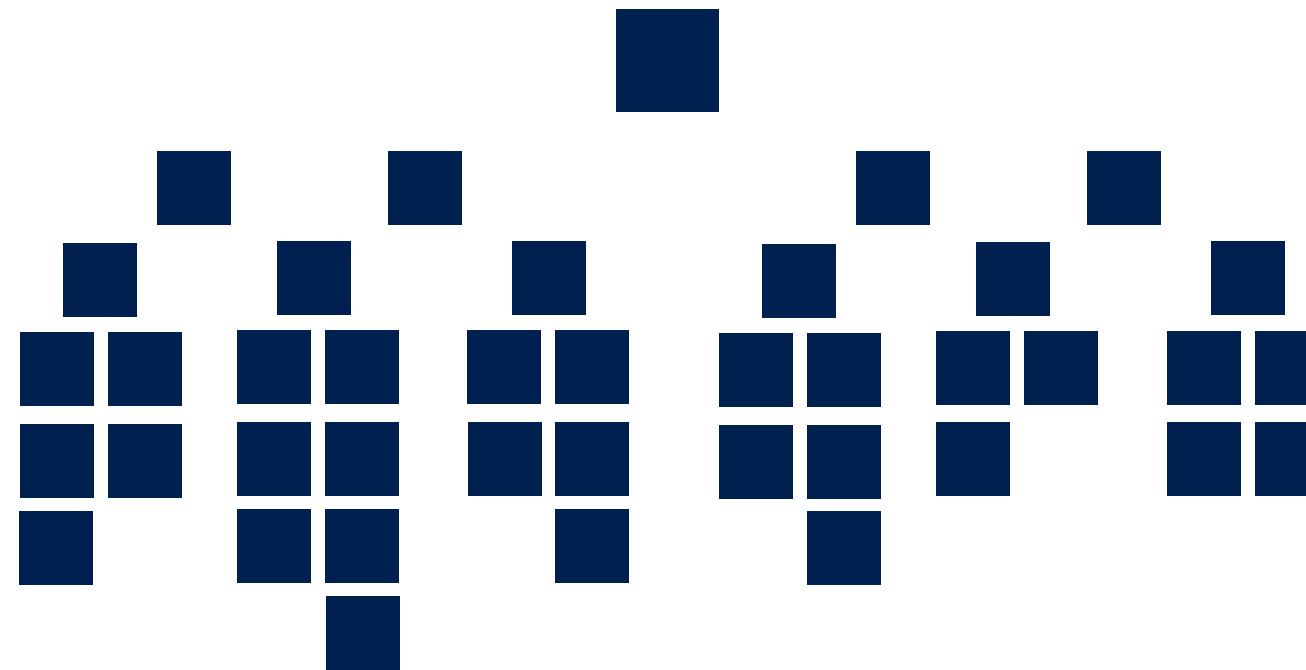


## Roles

# Program Management

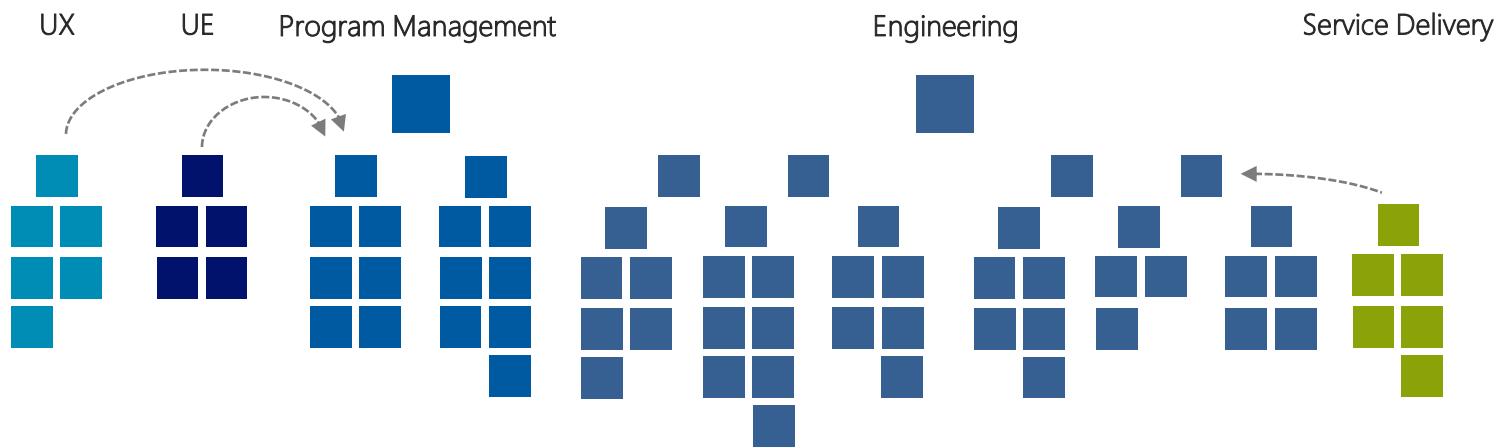


## Engineering



# Team Autonomy & Enterprise Alignment

# Yes, there are other roles...

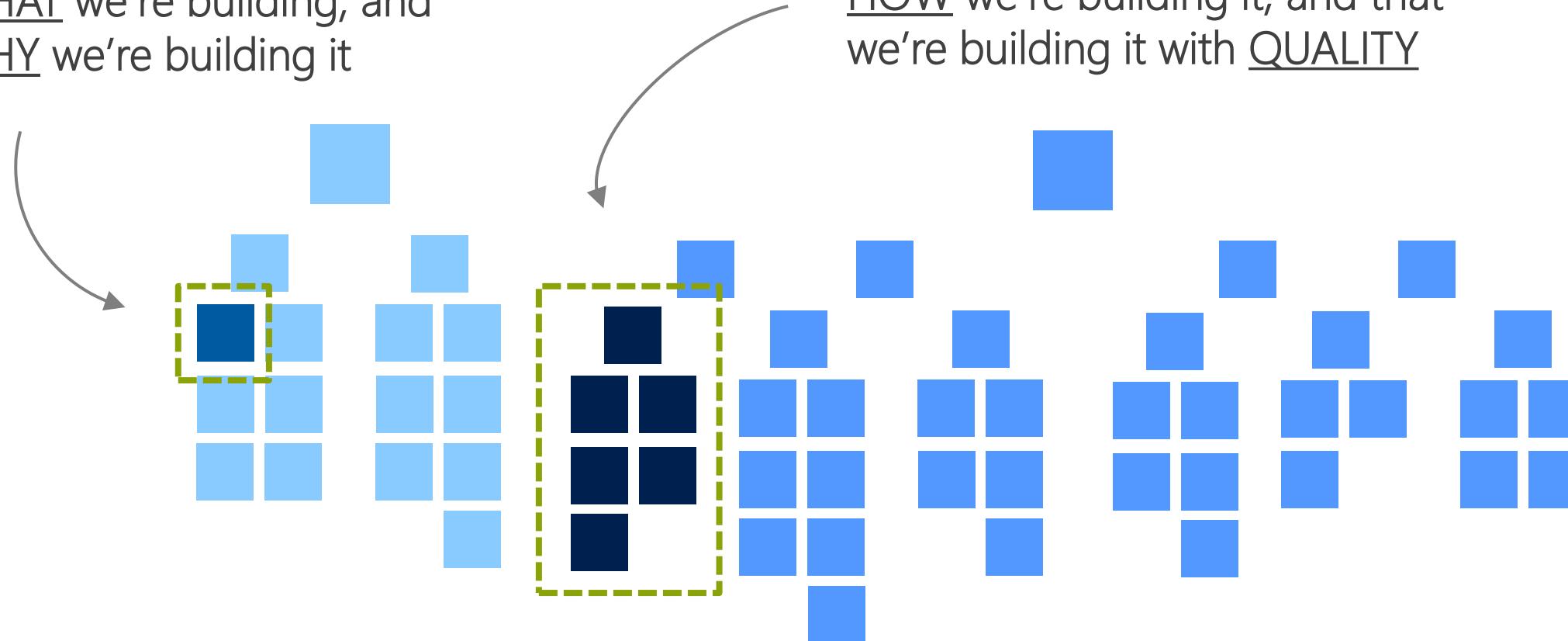


Service Delivery is integrated directly into our organization.

# Teams

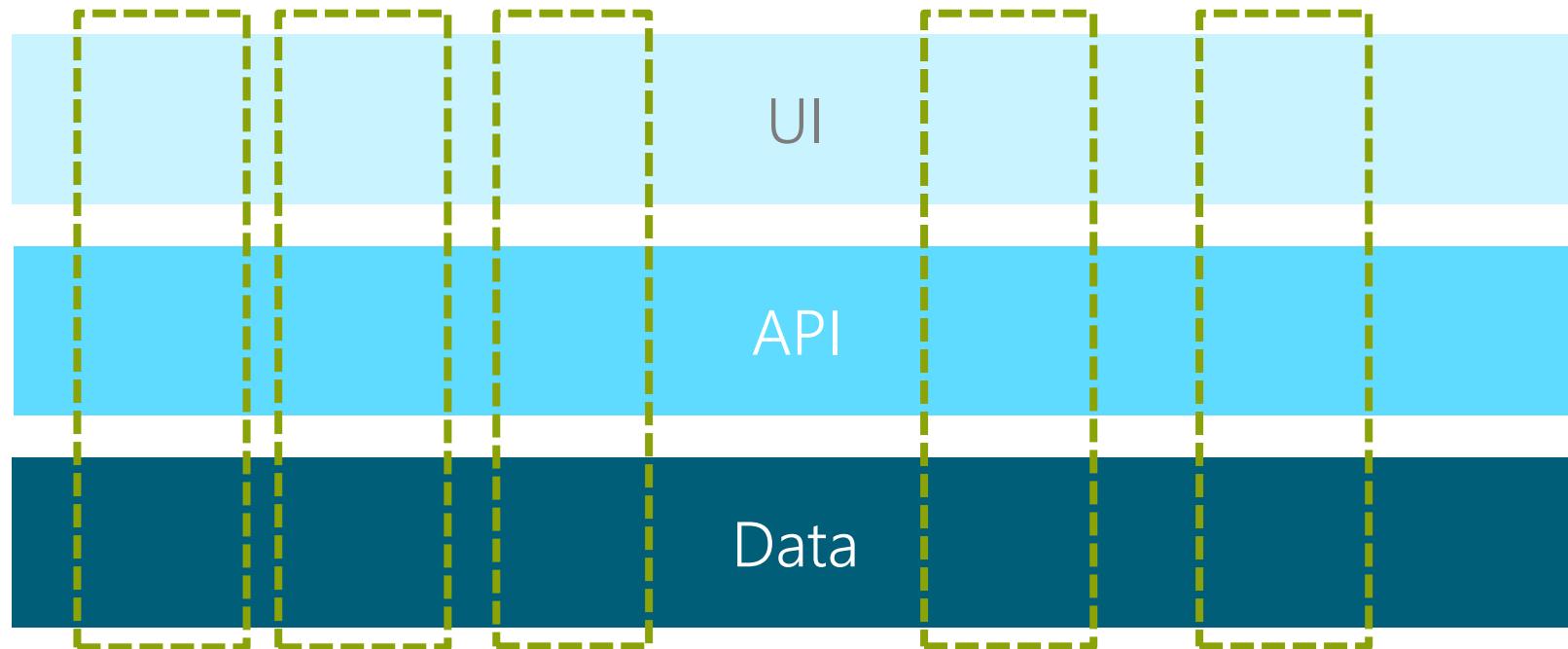
Program Management is responsible for:  
WHAT we're building, and  
WHY we're building it

Engineering is responsible for  
HOW we're building it, and that  
we're building it with QUALITY

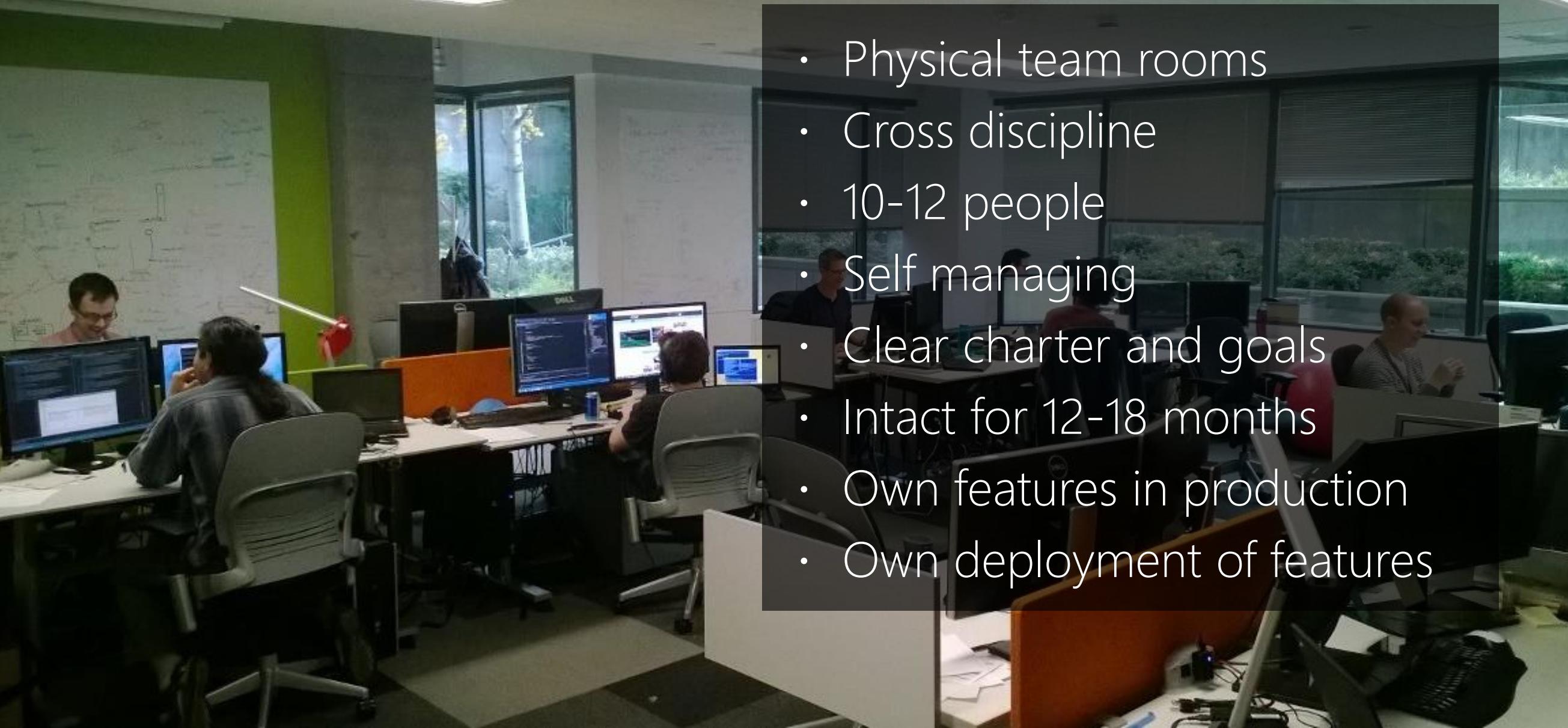


# Team owns their features

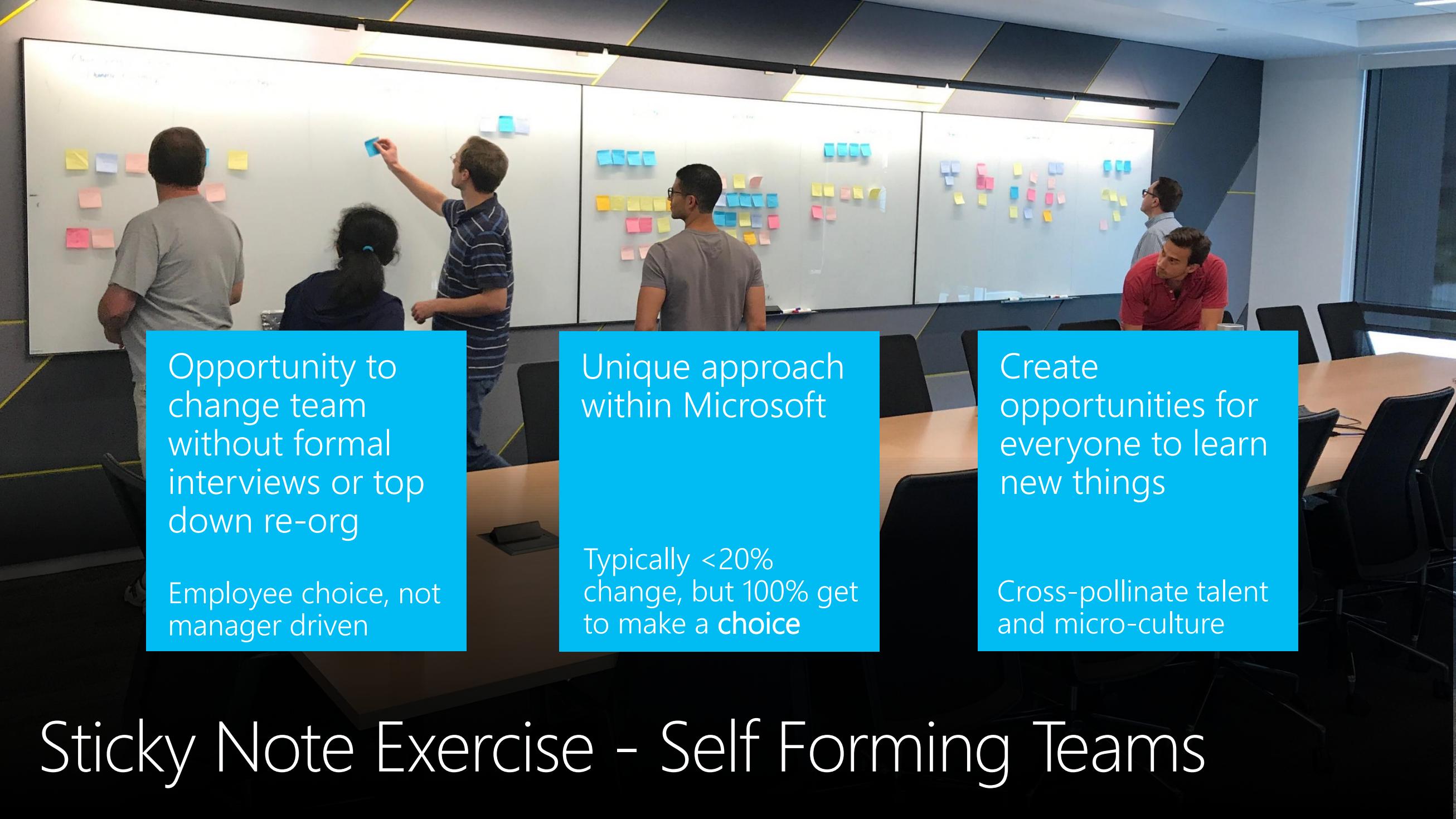
Teams own a feature all the way thru the stack to production



# 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

A photograph of a group of people in a conference room. They are standing around a whiteboard covered with numerous colorful sticky notes. One person is reaching up to place a blue sticky note. The room has a modern design with a large window on the right. In the foreground, there is a dark overlay with three blue boxes containing text.

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

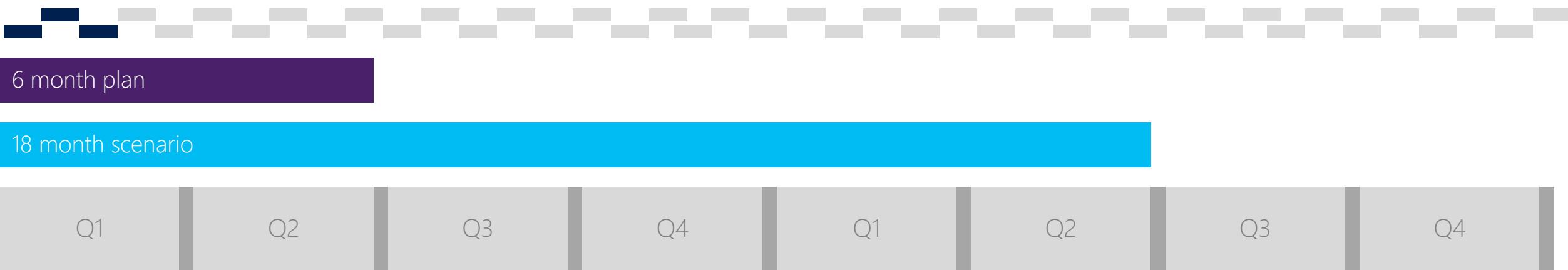


Leadership is responsible  
for driving the big picture

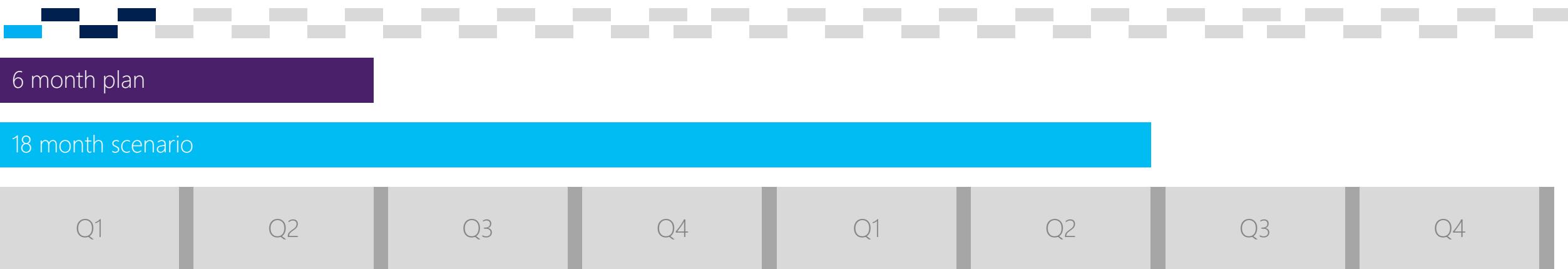
Teams are responsible for the detail



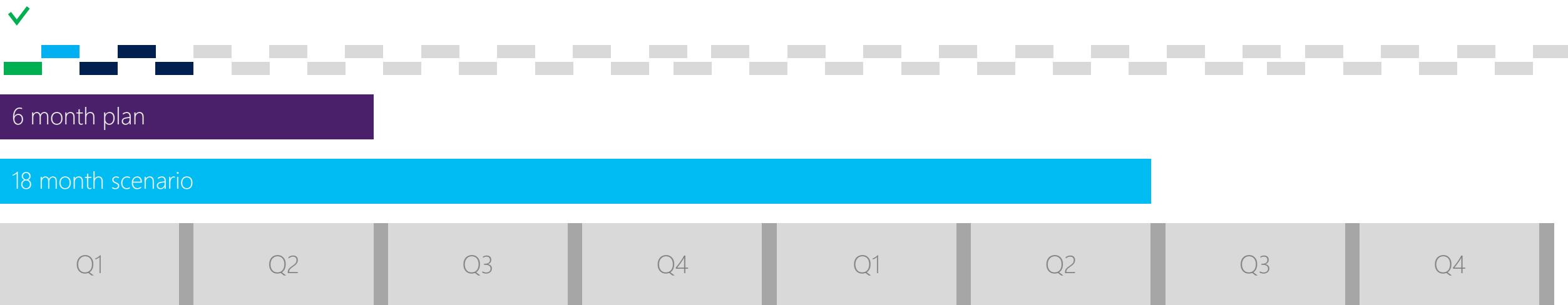
# Planning



# Planning

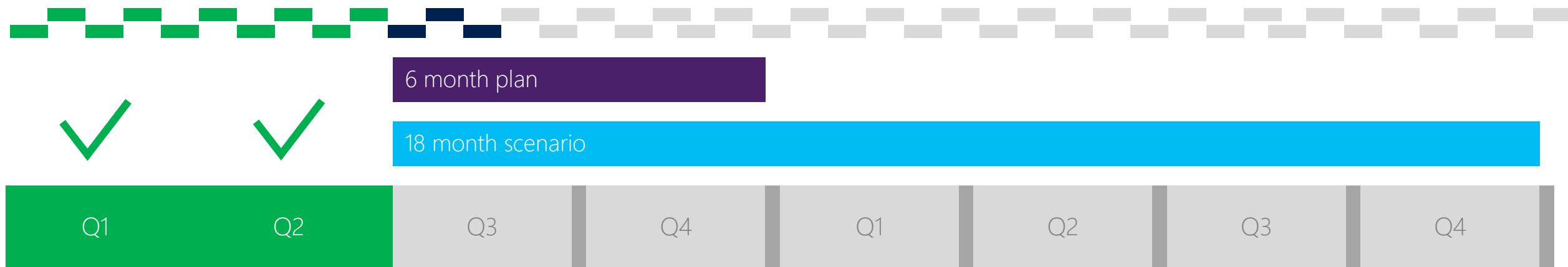


# Planning



# Planning

Remember: A plan is not useful. Planning is.



# 3 week sprints

The sprint plan



Week 1

Week 2

Week 3

Progressive Deployment

Sprint 97

Week 2

Week 3

Sprint 98

Week 1

Sprint 99

Week 2

Week 3

What we accomplished



930!



# Sprint mails

Sandeep Chadda | VSCS Sprint Communication; VS.in Core Team; VS.in Agile Experience FTE +  
Agile X: Sprint 97 Summary and Sprint 98 Plan

3/25/2016

**Agile X**  
Sprint 97 Summary  
Sprint 98 Plan

**Links:**  
[Backlog](#)

## Sprint 97 Summary

The sprint summary covers:

- a. Live Updates
- b. Board filtering
- c. Context menu enhancements
- d. Annotation Infrastructure for Test annotations

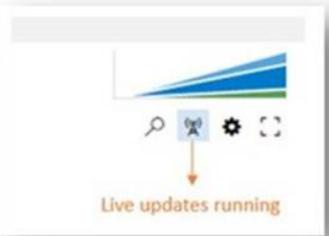
## Sprint Video:

[LiveUpdates.wmv](#) (~5min)

[FEATURE\\_486892: Live Updates \(a.k.a Auto board refresh\)- user voice](#)

- Live updates reflects stack rank related changes on the board
- Events are batched while sending to clients to optimize performance and load.
- Manage live updates when users go offline or when the tab on which the Kanban board is open is inactive for a long duration
- Manage live updates as a user level setting

The feature is going to be rolled out to Stage0 post M97 deployment. Post enabling the feature, board users will have to turn ON live updates on their boards.



[FEATURE\\_486893: Board Filtering - user voice 1, 2, 3, 4](#)

- Filtering Kanban board based on Assignments, Iterations, and Work items ([Spec](#)). This feature is work in progress and will be shipped as part of M98 deployment.

[FEATURE\\_527945: Context menu enhancements](#)

- Users will be able to open the card by clicking on the context menu.

## Sprint 98 Plan

[FEATURE\\_486892: Live Updates \(a.k.a Auto board refresh\)- user voice](#)

- Stage wise rollout of Live updates
- Enable hotspots for Live updates
- Monitor performance and scalability

[FEATURE\\_486893: Board Filtering - user voice 1, 2, 3, 4](#)

- Complete assignment, work item, and iteration based filtering
- Enable hotspots for filtering
- Start working on tag based filtering

[FEATURE\\_517978: Checklist standardization](#)

- Click on title to open work item
- Add context menu to checklists to open and edit work item
- Remove double click experience

## CodeLens

- Stand up India SU
- Finish Key vault settings migration for Test code

Thanks,  
AgileX Team| VS.in



Aakash Johari



Giridharan Narayanan



Prashanth Swaminathan



Ronak Agarwal



Rosen Dash



Sandeep Chauhan



Saranya Baskaran



Sravanth Bangari



JP



Sandeep Chadda



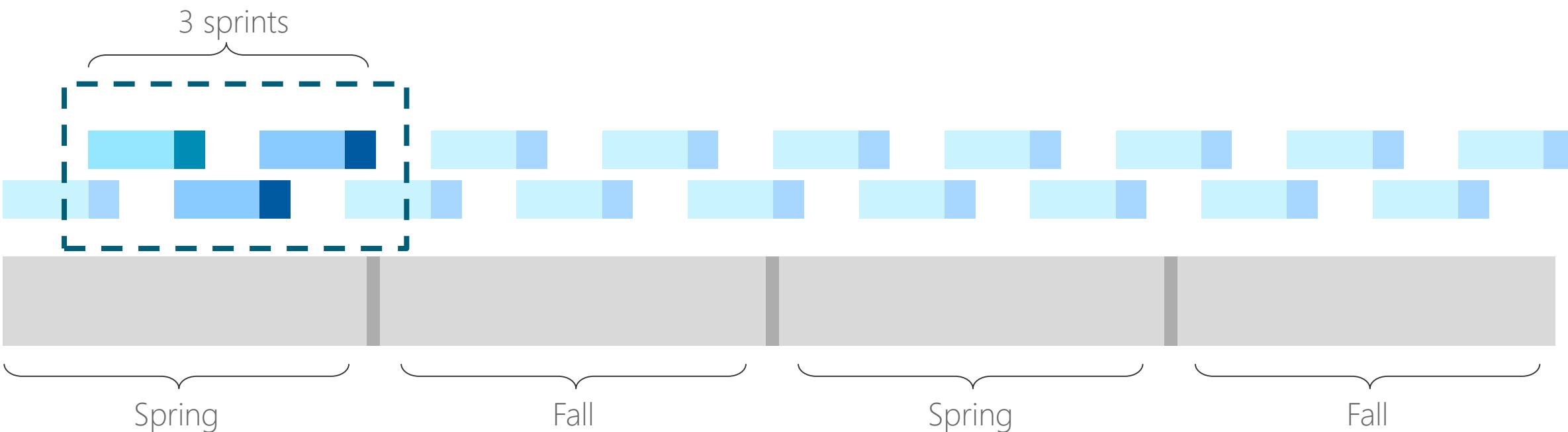
Sulabh Upadhyay



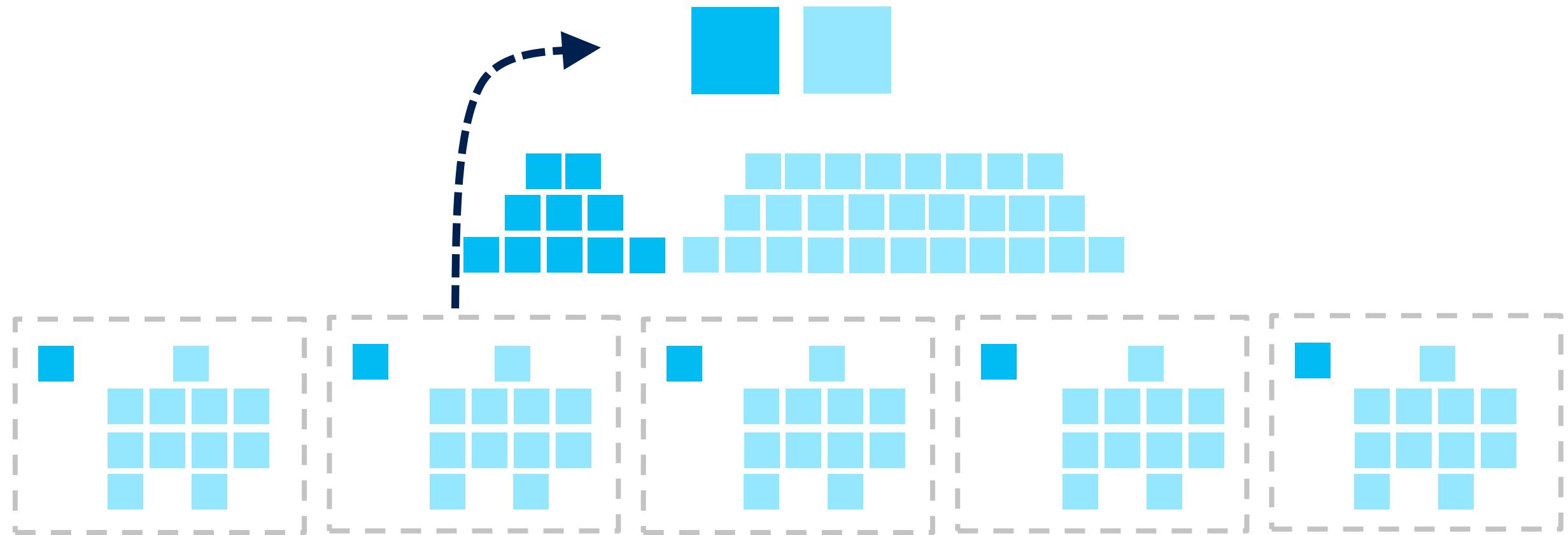
Akash Saxena

# Team Chats

- What's next on your backlog?  
What's your debt situation?  
Any issues? Dependencies?

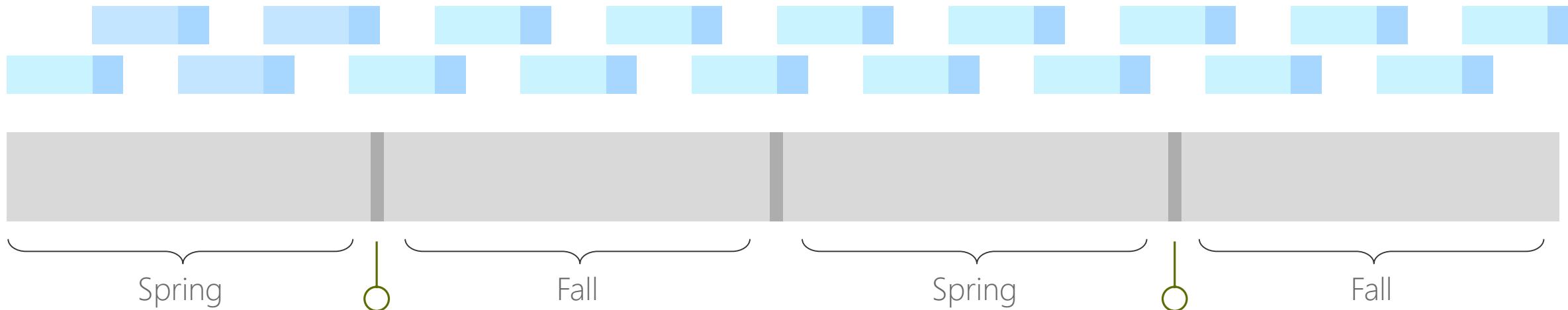


# Direct. No “lost in translation”.



# Experience Reviews

- Storyboard of customer experience
- High level execution plan – sprints, !hours
- Feedback, feedback, feedback



# Benefits

- 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 we need to change course

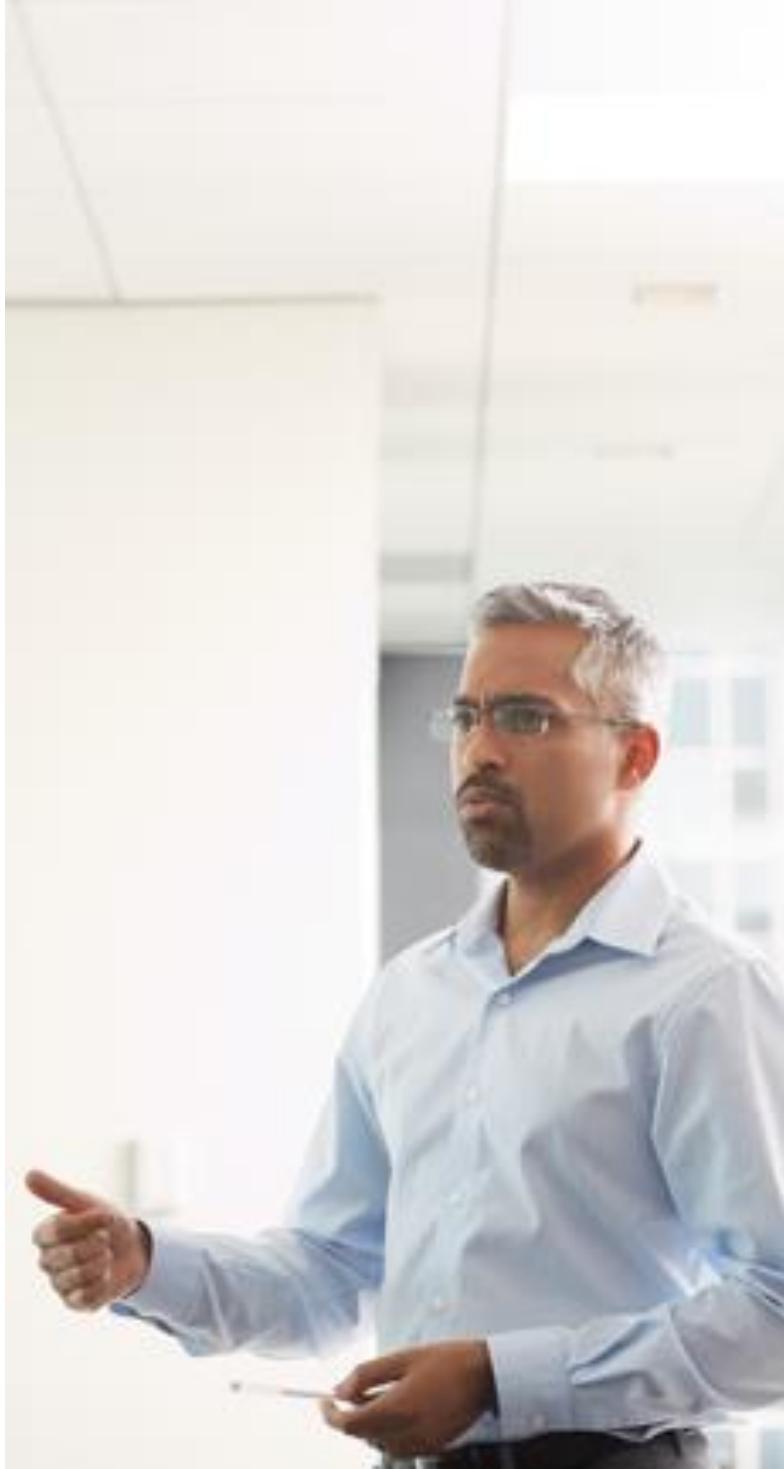
# EPIPHANY #3

Cross-functional teams who own and take responsibility for features from ideation to deprecation, are the ones that collaborate and delight their customers the most.

They are also represented in 2AM calls the least ☺



# Five habits we've learned so far



# Bug Cap

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

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



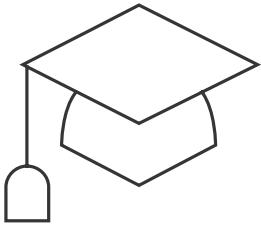
# Bug Cap

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

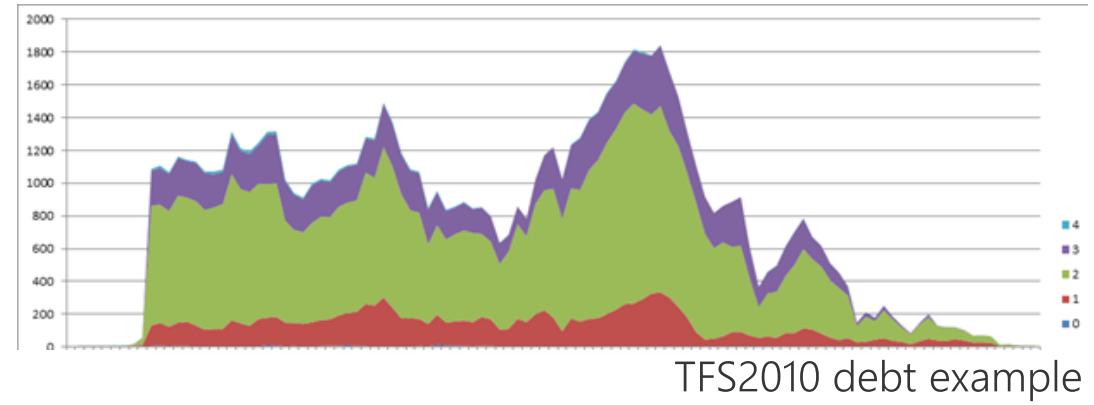
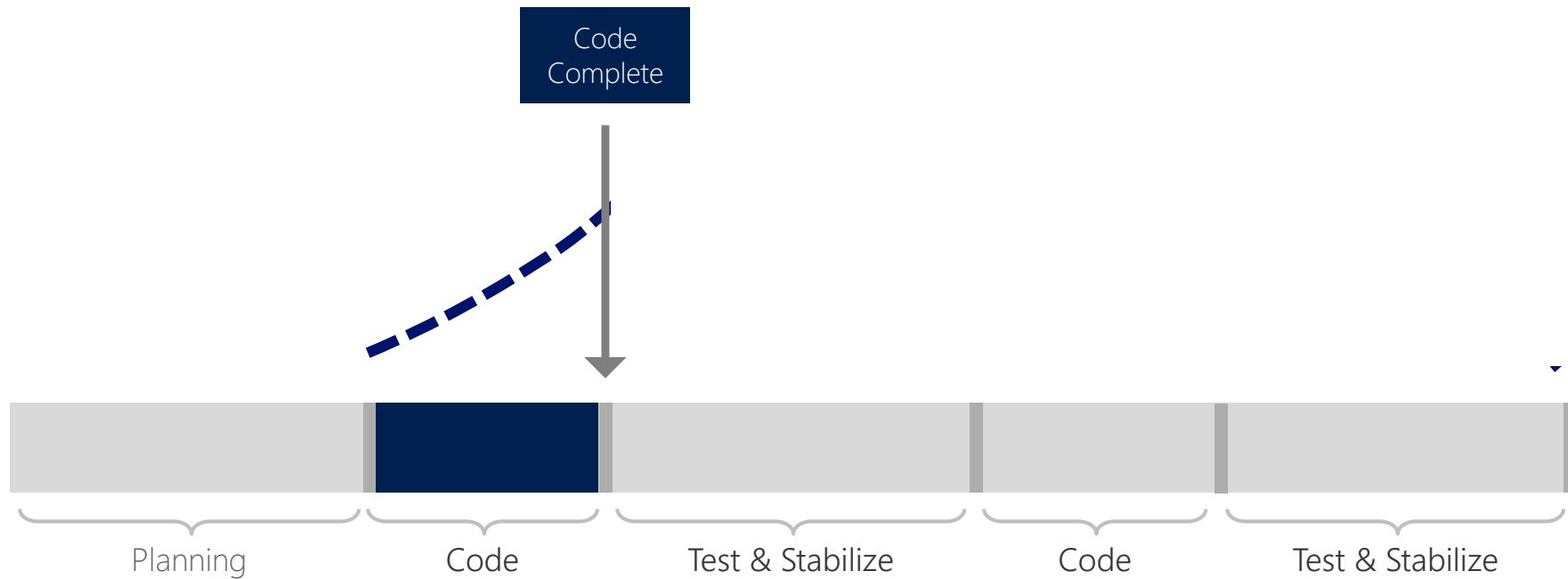
$$10 \times 5 = 50$$

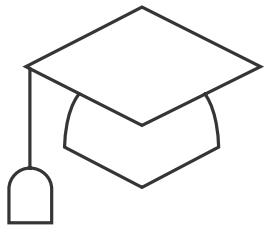
Rule: If your bug count exceeds your bug cap... stop working on new features until you're back under the cap.



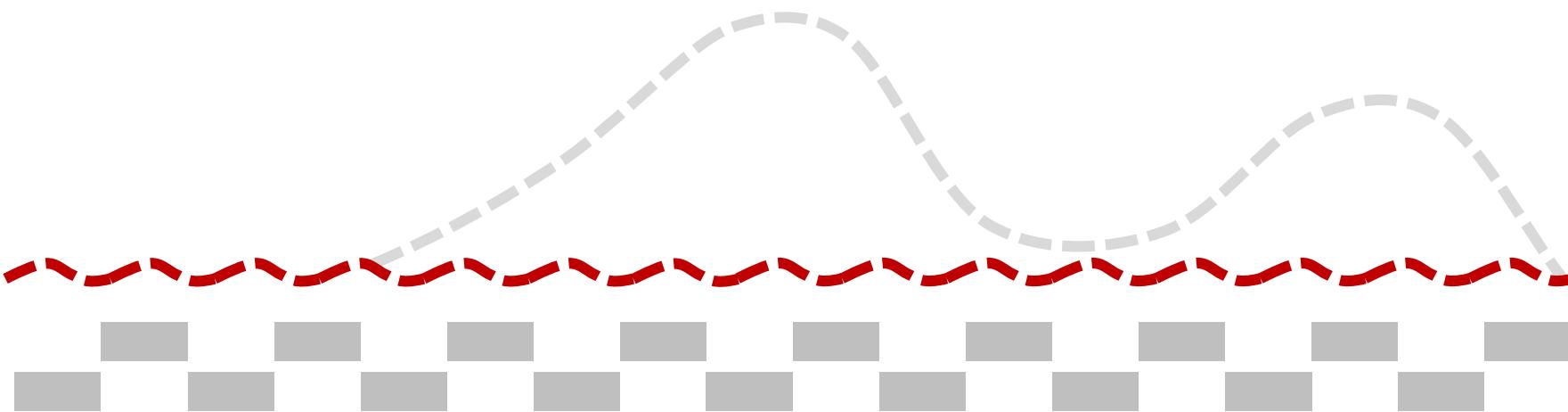


# Before – Debt Cycle





After



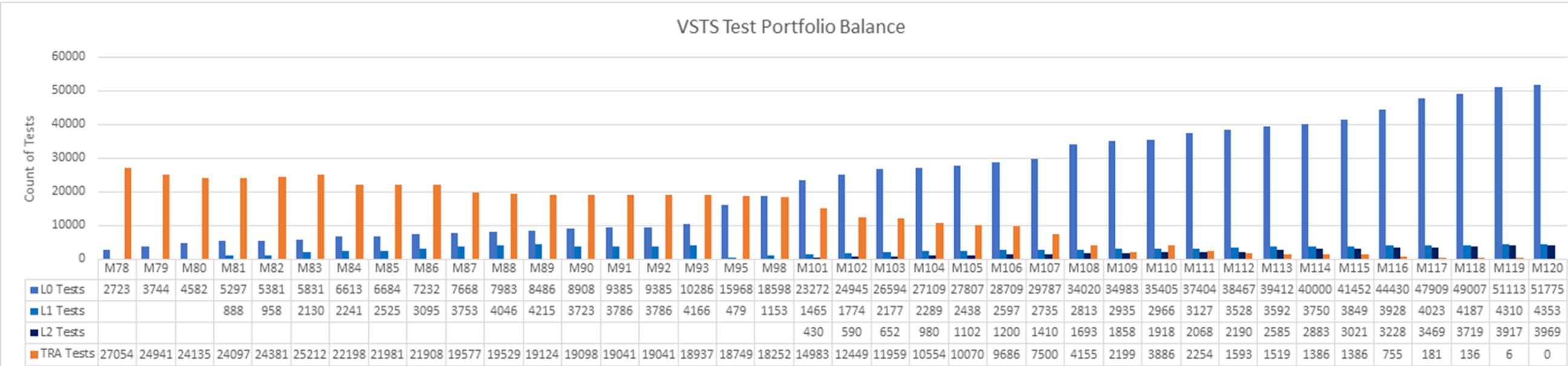
# Test Portfolio - Shift Left from Integration to Unit

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 a Microsoft VSTS pull request page for pull request 260803. The pull request has been completed by Karthik Balasubramanian on 23/09/2017 at 11:45, merging changes from 'users/kabalas/addshortcuts' into the 'master' branch. The status bar indicates 'All resolved'. Below the status, there are tabs for Overview, Files, Updates, Commits, and Conflicts. The 'Overview' tab is selected.

**Policies**

- Required:
  - ✓ 1 reviewer approved
  - ✓ Build succeeded
  - ✓ CredScan Validation succeeded
- Optional:
  - ✓ All comments resolved
  - Artifact Services
  - Packaging & Signatures
  - Tfs.SelfHost Set
  - Tfs.SelfHost Set

**Test Results**

Total tests	Failed tests	Pass percentage
71283 (+71283)	0 (+0)	100% (+100%)
Passed (71283)	New (0)	
Failed (0)	Existing (0)	
Others (0)		

**Run duration**: Not Reported  
6m 35s (+6m 35s)

**Reviewers**

- WIT IQ via Matthew Manela
- Matthew Manela Approved
- Modern WIT
- WIT PI

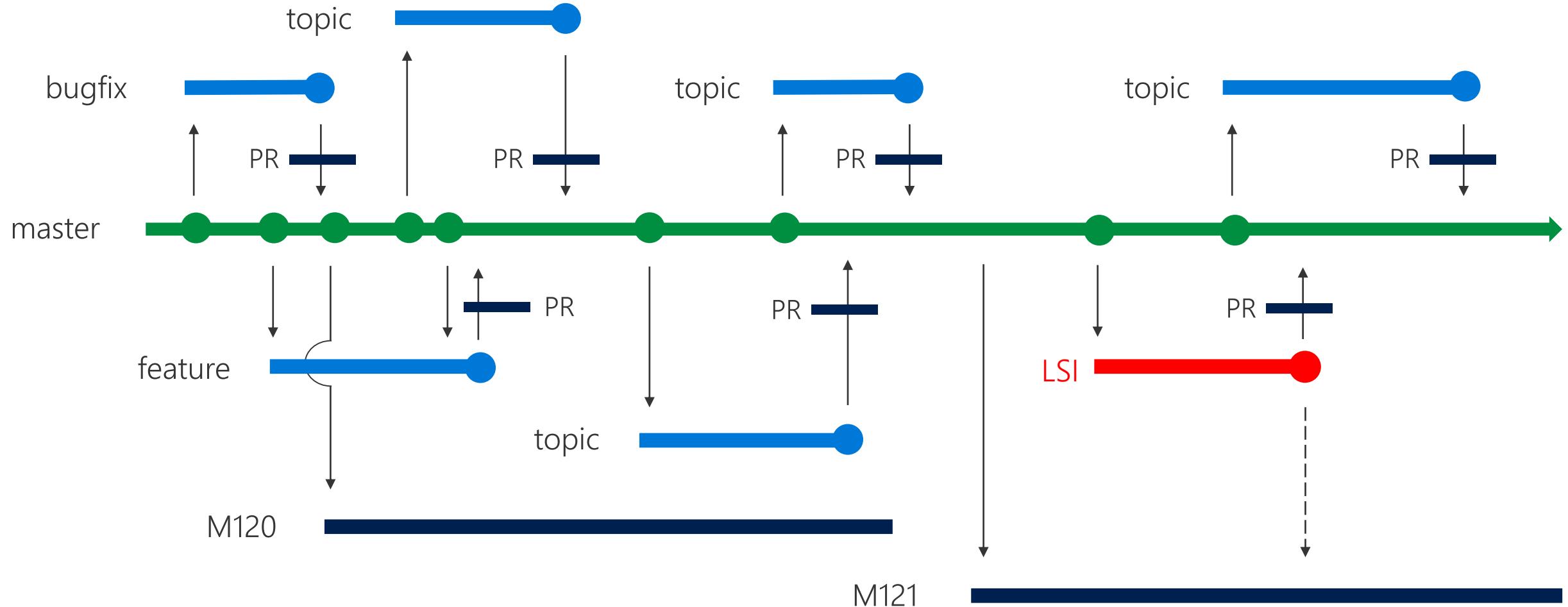
**Labels**

Add label

**Comments**

- Matthew Manela 23/09/2017 I think we can just use hub navigation service and do it. Just look how the menu code does it. Shouldnt be hard. @Karthik Balasubramanian
- Matthew Manela 23/09/2017 And we should use XHR
- Matthew Manela joined as a reviewer
- Karthik Balasubramanian 23/09/2017 pushed 1 commit creating update 6
- Karthik Balasubramanian 23/09/2017 15:14 Reverting method override

# Release Flow Branching Structure



# EPIPHANY #4

Automation and performance are key to convince engineering teams to embrace SHIFT LEFT practices.

Once engineers see the value and realize that greater quality results in less 2AM calls they will evangelize the practice for you.

It's all about the 2AM call!



# Five habits we've learned so far



# Architecture modernization

Move to cloud

- Less heat and more leg room under table
- More processing power, flexibility, and scalability

VSTS

- Service oriented architecture
- Move to micro services, stateless services, and containers
- Raise redundancy and scalability, and lower cost

# DevOps isn't magic



# 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
- 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
- Zero debt
- Mockups in PPT
- Inner source
- Flattened organization hierarchy
- User satisfaction determines success
- Features shipped every sprint

# Lastly the DevOps Ranger Transformation

## Before

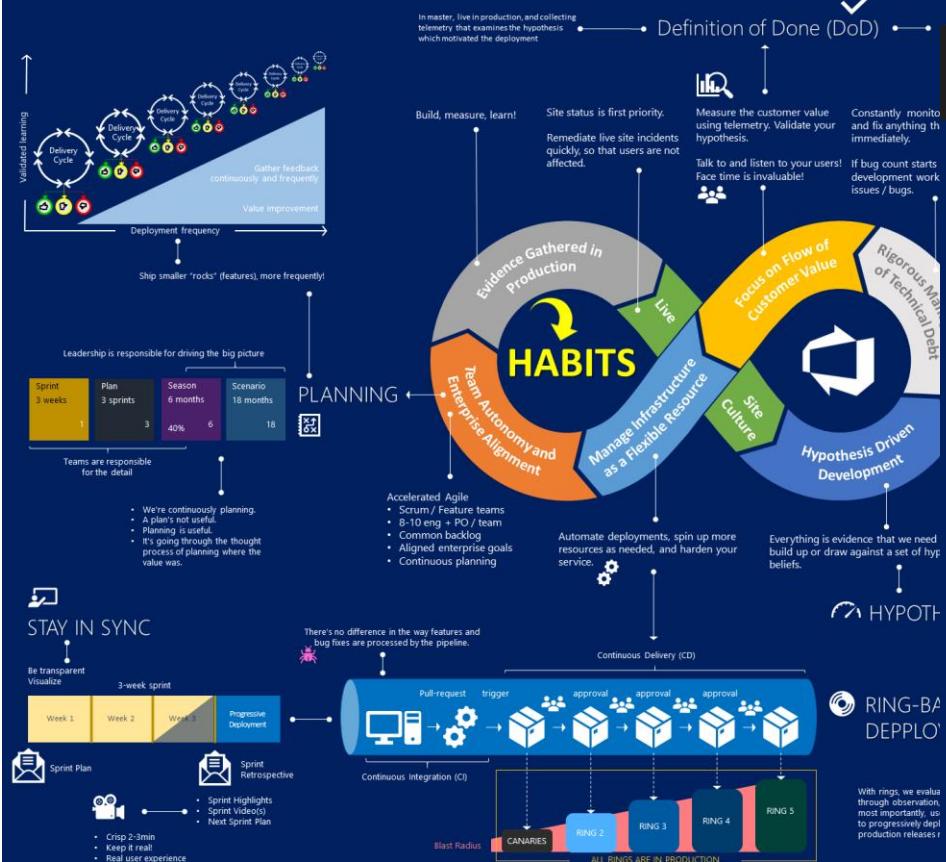
- 10-15 person teams
- 2 program managers
- Manual and error prone builds
- Manual and error prone releases
- 6-12 sprint cadence
- 1 month sprints
- Issues detected by users
- Days to weeks to resolve issues
- Hours to build
- Days to release

## After

- 2-5 person teams
- 0.25 program managers
- Automated CI
- Automated CD
- 3-5 sprints cadence
- 3-week sprints
- Proactive telemetry
- Minutes to days to resolve issues
- Minutes to build
- Minutes to release

# DevOps Habits @ Microsoft

DevOps is the union of people, process, and products to enable continuous delivery of value to our end users. - @donovanbrown



DevOps at Microsoft  
aka.ms/DevOps  
DevOps Lab  
aka.ms/DevOpsLab  
Whitesource OSS Security  
www.whitesource.com

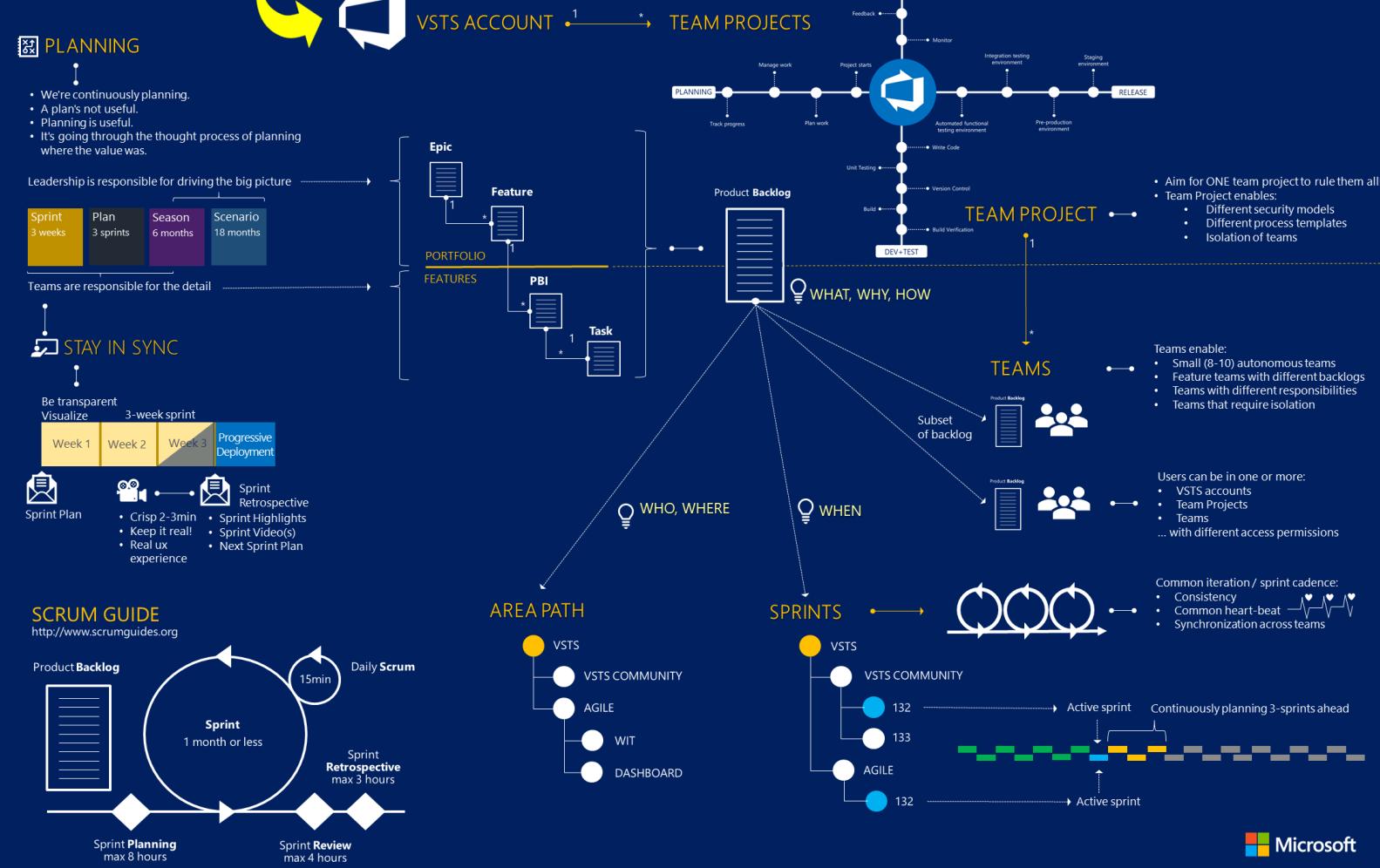
Welcome to DevOps  
aka.ms/WhatIsDevOps  
DevOps Hands-on Labs  
aka.ms/DevOpsLabs  
LaunchDarkly Flags  
www.launchdarkly.com

DevOps Technical Articles  
aka.ms/DevOpsArticles  
Visual Studio Team Services  
www.twitter.com/vsts  
2018.03.09 v1.1  
@vsts @almrangers

## RELEASE FLOW BRANCHING STRUCTURE



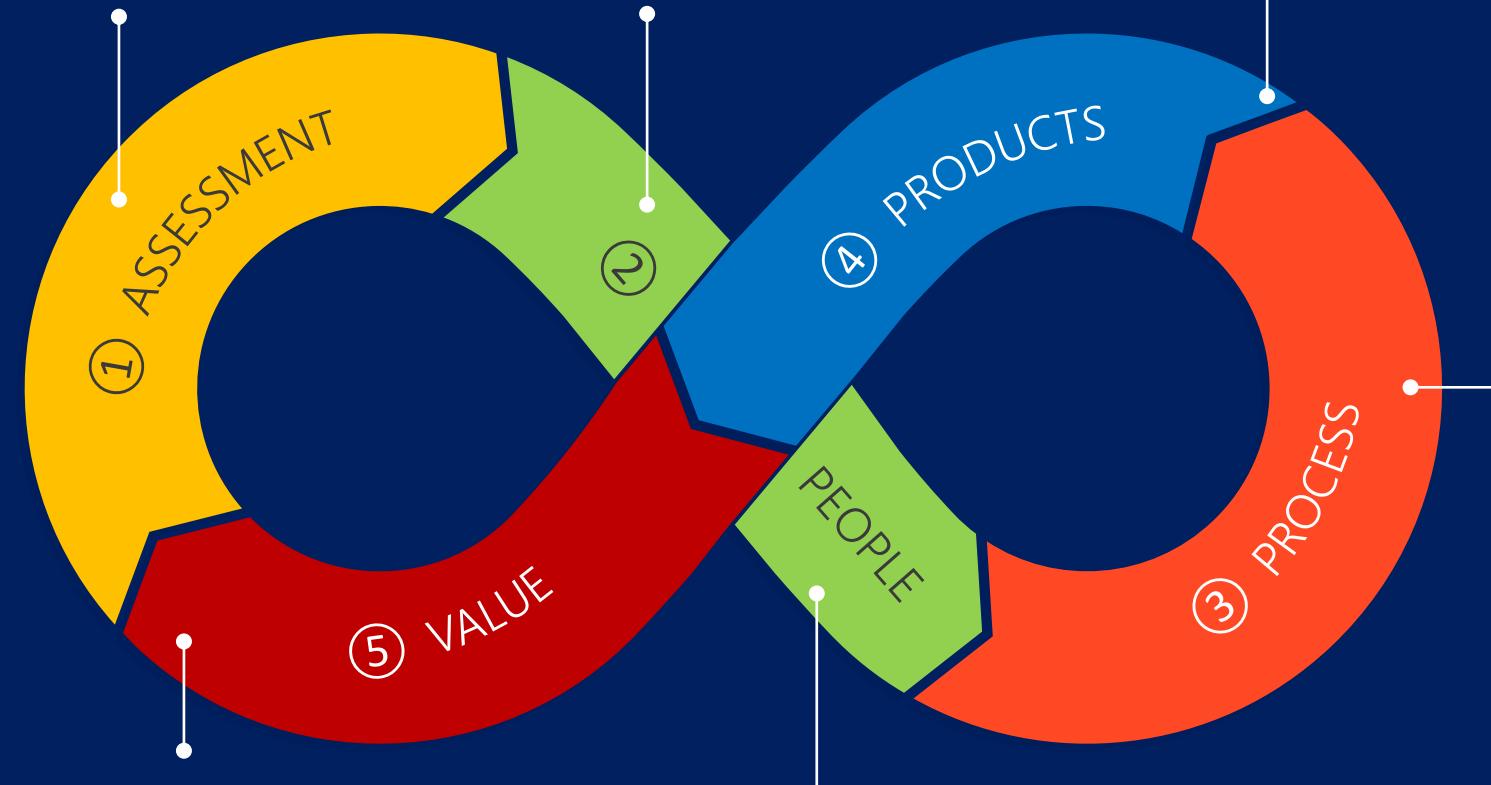
# Planning with VSTS @ Microsoft



# DevOps Getting Started

DevOps X-Ray Assessment  
<https://www.devops-survey.com>

Microsoft DevOps Self-Assessment  
<https://aka.ms/devopsassessment>



## Key performance indicators

- Change failure rate
- Deployment frequency
- Lead time for changes
- Time to recover

## Leadership

- Accept failures as normal
- Inspirational communications
- Intellectual stimulation
- Own the vision
- Personal recognition
- Supportive leadership

## Culture

- Climate of learning
- Cross-functional collaboration
- Dogfooding and canary environments
- Effective use of tools
- Everyone empowered
- Everyone responsible and on call
- Trained to run the business
- Value stream-based teams

Transforming the culture is the biggest challenge.

All **PEOPLE** need to buy into the transformation, be kept up to date on all initiatives, understand how their roles will be affected, collaborate transparently, and take responsibility for their features.

**PRODUCTS** and technology are enablers, allowing teams and architects to focus on outcomes.

## Key **PROCESS** Goals

- Automate everything – fast, stable, consistent
- Celebrate success as a team and organization!
- Focus on quality (security, test, deploy,...)
- Lightweight change management process
- Loosely coupled architectures enable scaling
- Multiple releases per day

## Live Site Incidents

- Track live site incidents
- Remediate at root cause level

## Lean Management

- Create a production feedback loop
- Lightweight (empower, trust) change approvals
- Limit work in progress (WIP)
- Visual (dashboards) work management

## Lean Development

- Experiment, learn, and influence
- Gather & action feedback
- Make flow of work visible (dashboards)
- Work in small batches (WIP)



## HYPOTHESIS

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

## Continuous Delivery

- Built-in quality
- Continuous improvement
- Everyone is responsible
- Use computer for repetitive tasks
- Work in small batches

# Reference Information

A Rough Path – Brian Harry  
<https://aka.ms/bh-ff-sos>

503 Error – ALM | DevOps Rangers  
<https://aka.ms/vsar-ff-sos>

Cost of feature flags  
<https://t.co/5xulA1hYI3>

DevOps @ Microsoft  
<https://aka.ms/devops>

ALM | DevOps Ranger Community Articles  
<https://aka.ms/vsarpublications>

This deck and others  
<https://tinyurl.com/wpschaub>



# Thank you!

AJATO TRANSFORMATIONS LTD.

EMAIL	wpschaub@outlook.com
TWITTER	@wpschaub
BLOG	willys-cave.ghost.io
GITHUB	github.com/wpschaub
OSS ARTICLES	opensource.com/users/wpschaub

VICTORIS DEVOPS MEETUP