

Visual Studio Online: Journey to DevOps

Thoughts to share from Microsoft Developer Division's transformation to DevOps

Sam Guckenheimer

Product Owner, Visual Studio Product Line
Microsoft Corporation

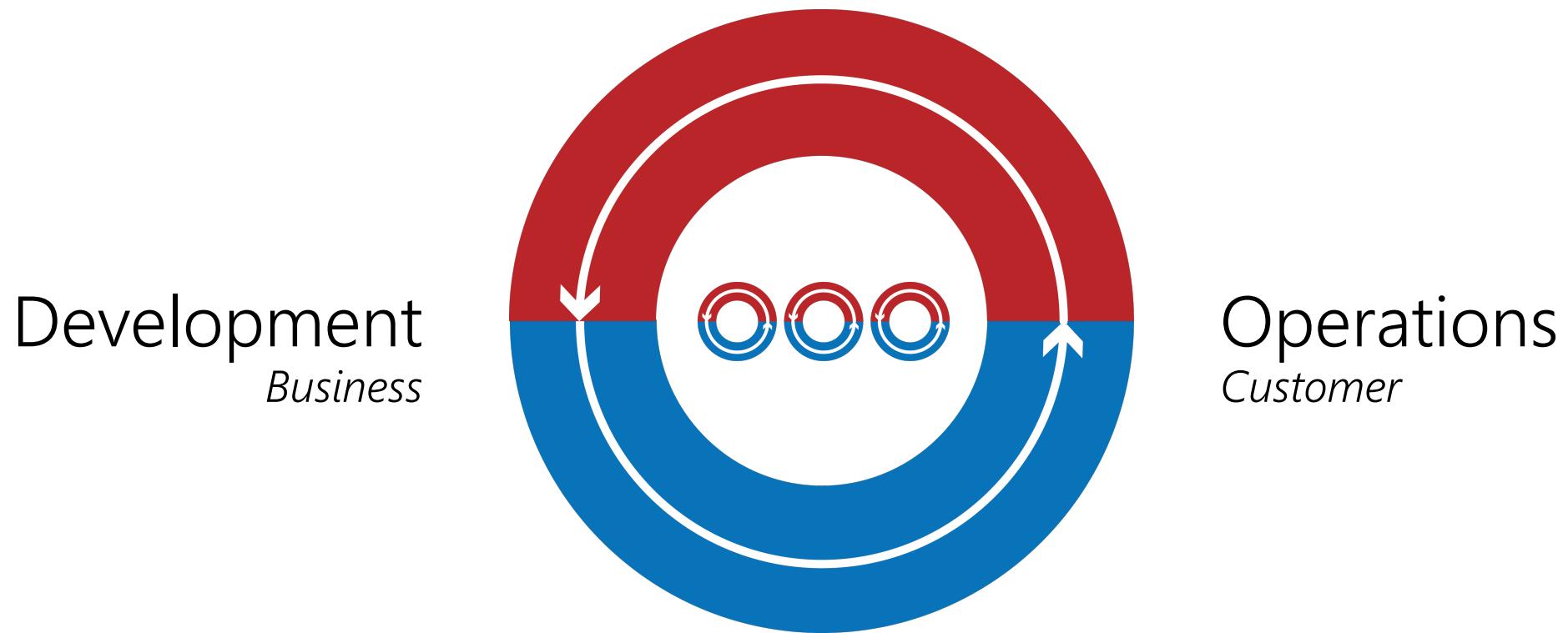
@samguckenheimer

@shitsamgusays

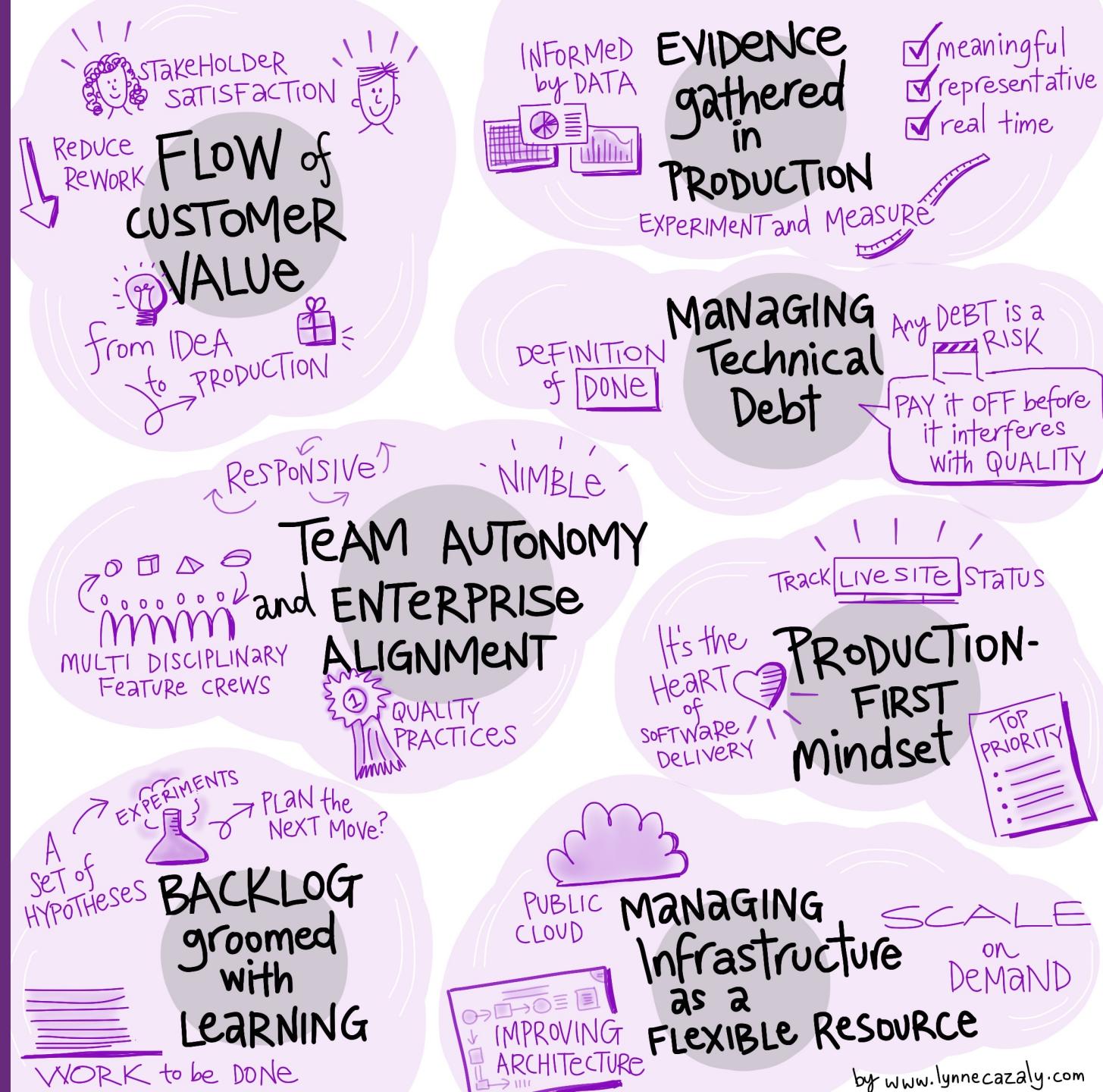
<http://aka.ms/devops>



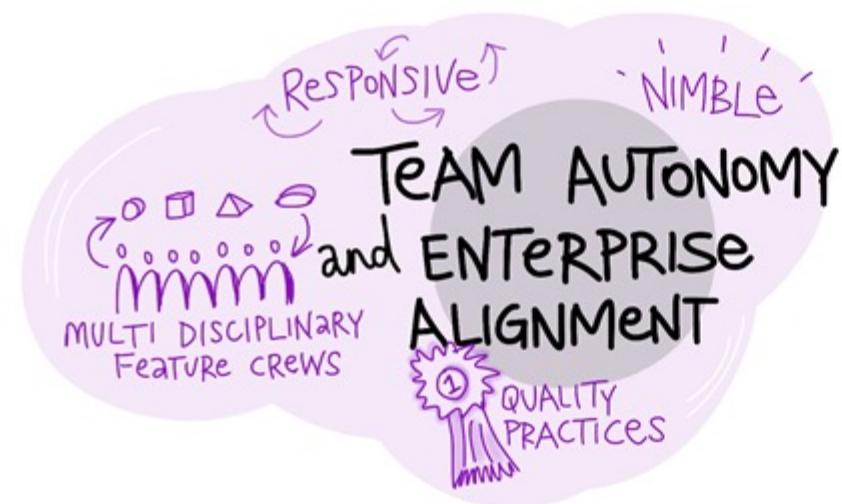
Canonical view of DevOps (aka the “Three Ways”)



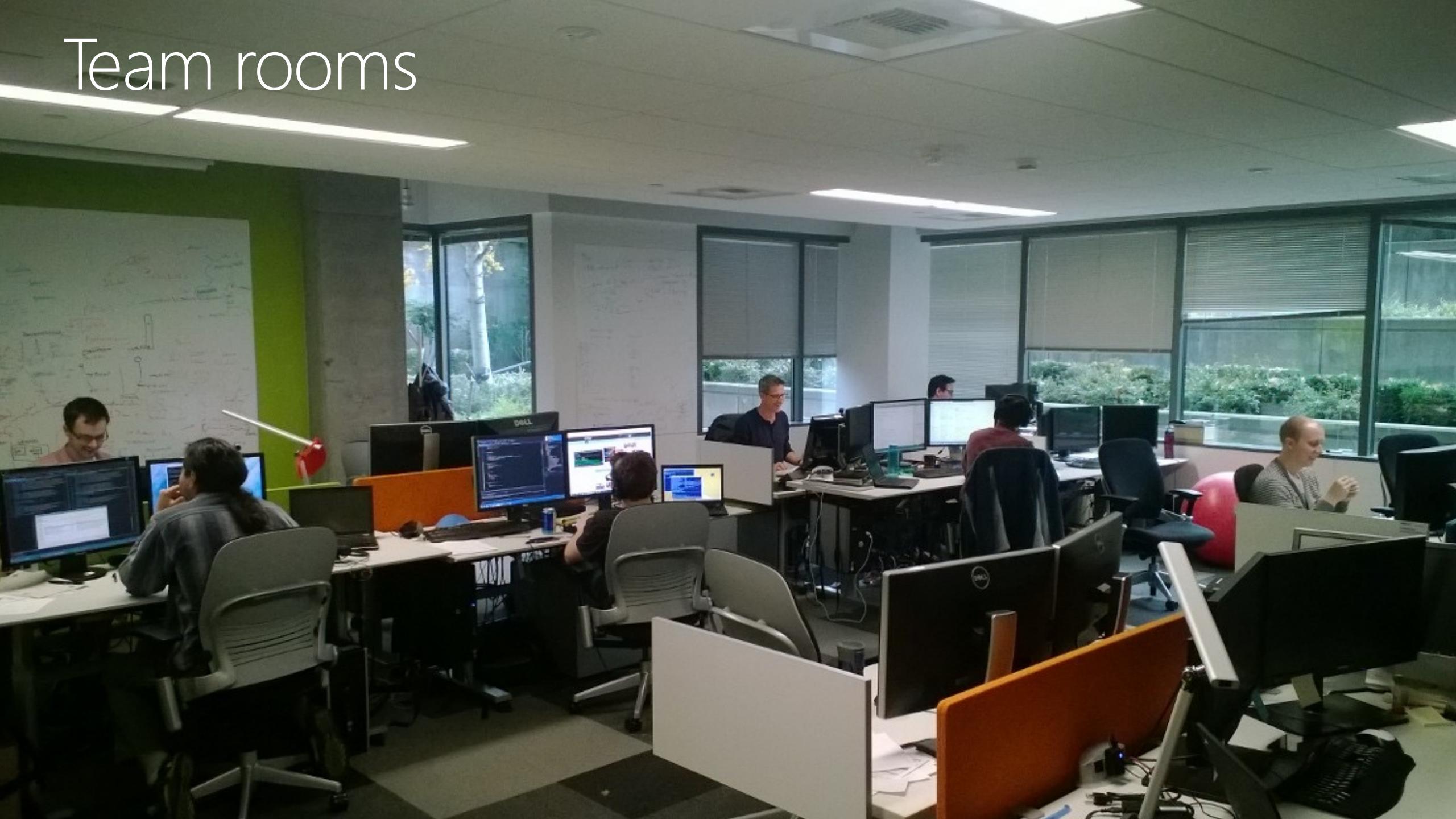
Our learnings as a SaaS provider



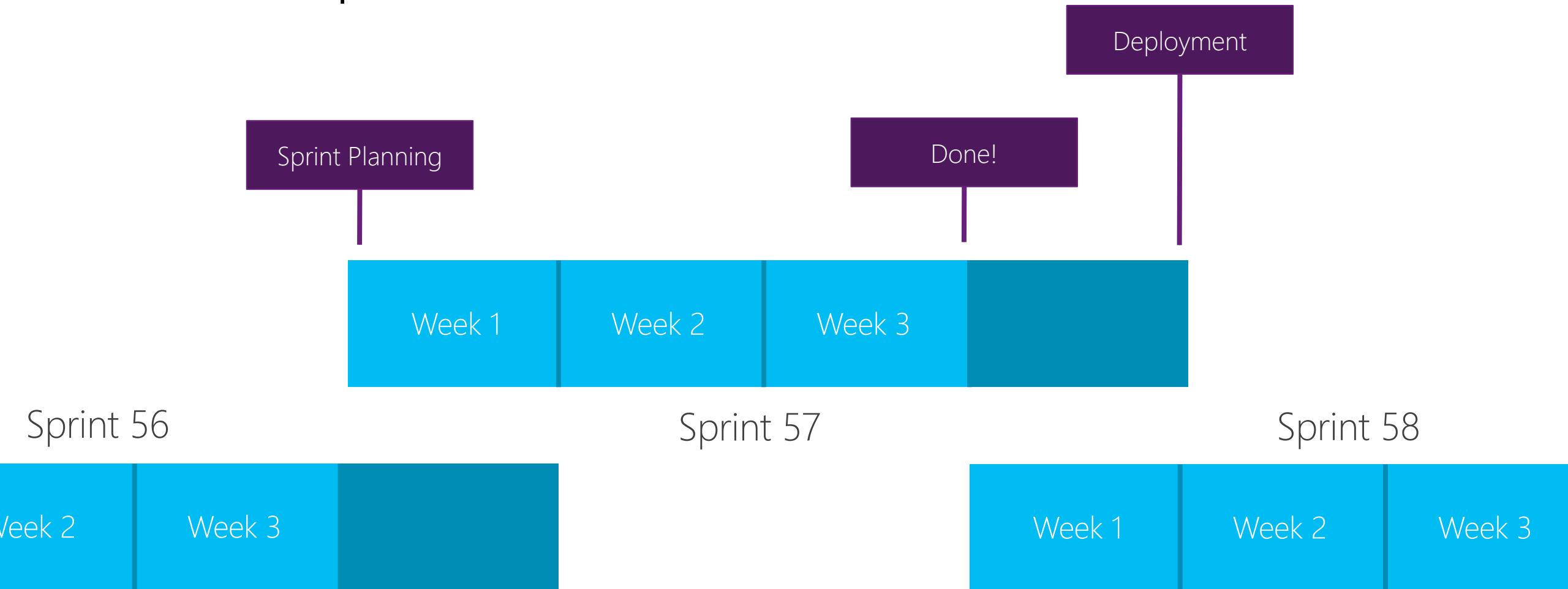
Our learnings as a SaaS provider



Team rooms



3 week sprints



Planning horizons



Sprint mails

Collaboration - Sprint 41 Plan - Message (HTML)

FILE MESSAGE

Mon 12/3/2012 12:42 PM

David Goren Elizondo

Collaboration - Sprint 41 Plan

To: VS Cloud Services All
Cc: TFS Collaboration

Retention Policy: Delete after 1 year (1 year) Expires: Never

Our team has started to collaborate with VC folks on the user story that will enable the E2E for GIT scenarios. We expect to be done with this E2E experience this sprint. We acknowledge the fact that we're entering 'holiday' sprints, but our goals will also include remaining under the bug cap, completing our engagement with the Airstream team, and continuing the 'Team room' momentum with a more formalized design after the spiking that happened last sprint.

Goals

- Remain under the bug cap
- Collaborate with the VC team on enabling the E2E for the GIT story
- Continue the 'Team Room' momentum with a more formalized design

User Stories

ID	Title	Goal for end of sprint
832408	Bugs (Starting sprint with 23)	Remain under the bug cap
987548	I can mention a work item (#ID) as part of a GIT commit	Enable, together with the VC team, the E2E story
983673	Airstream	Complete engagement with the Airstream team
1021717	I can understand our design approach for the virtual Team Room	Start on a more formalized design for Team Room

Thanks,
The Collaboration team

David Goren Elizondo RE: Can't edit HTML field in Excel - in some situations

Collaboration - Sprint 41 Results - Message (HTML)

FILE MESSAGE

Thu 1/3/2013 9:38 AM

Justin Marks

Collaboration - Sprint 41 Results

To: VS Cloud Services All
Cc: TFS Collaboration

i You forwarded this message on 1/7/2013 8:54 AM.

Even with the holidays the team was able to get a bunch of stories completed. The Collaboration team has really embraced collaborating with other feature teams:

- We've had a strong collaboration with the VC team on delivering an E2E story around linking GIT commits to work items as can be seen in the below sprint video
- We worked with the Airstream team to pull down and validate the RI so that the discussion service is now being built in production using Airstream
- With the help of the WIX team we added the ability to send email from the product backlog

All the while we continued to stay on top of bugs dropping below 20!

Sprint Demo

TFS Collaboration
Sprint 41 Demo

GIT work item association

Justin Marks RE: TFS 2012 - Excel integration: Hide "hidden" work item types

Our learnings as a SaaS provider

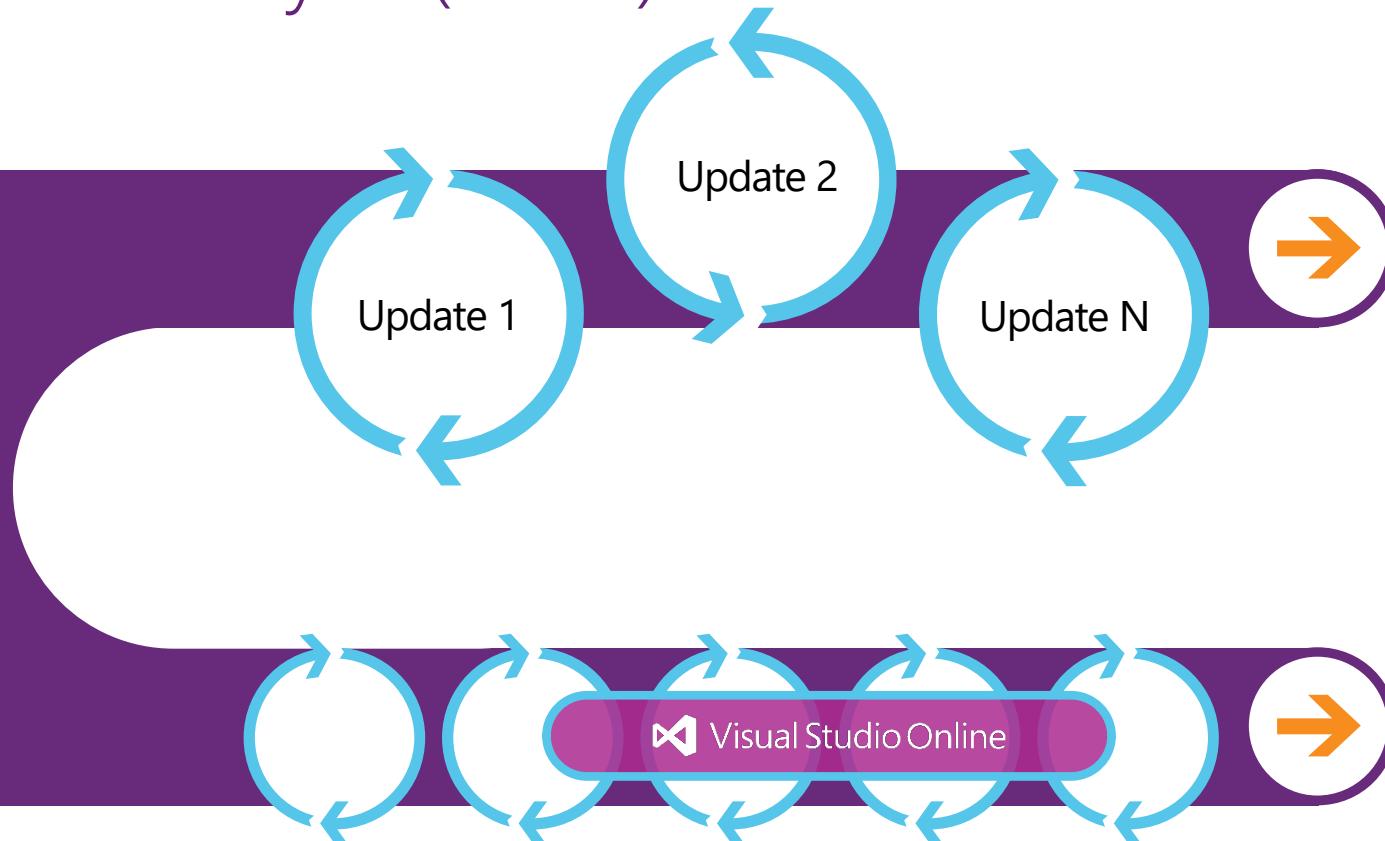


Code: Cloud first, then move on-premises

One code base with multiple delivery streams

Single master branch, multiple release branches

Shared abstraction layer (VSSF)



Goals: Stay clean and move fast

Ship frequently with minimum friction

Dream of a system that can do hands-free push of a hotfix to production in 15 minutes

Fast and predictable inner loop

Inner loop = Build>Deploy>Test cycle which runs reliably hundreds of times a day

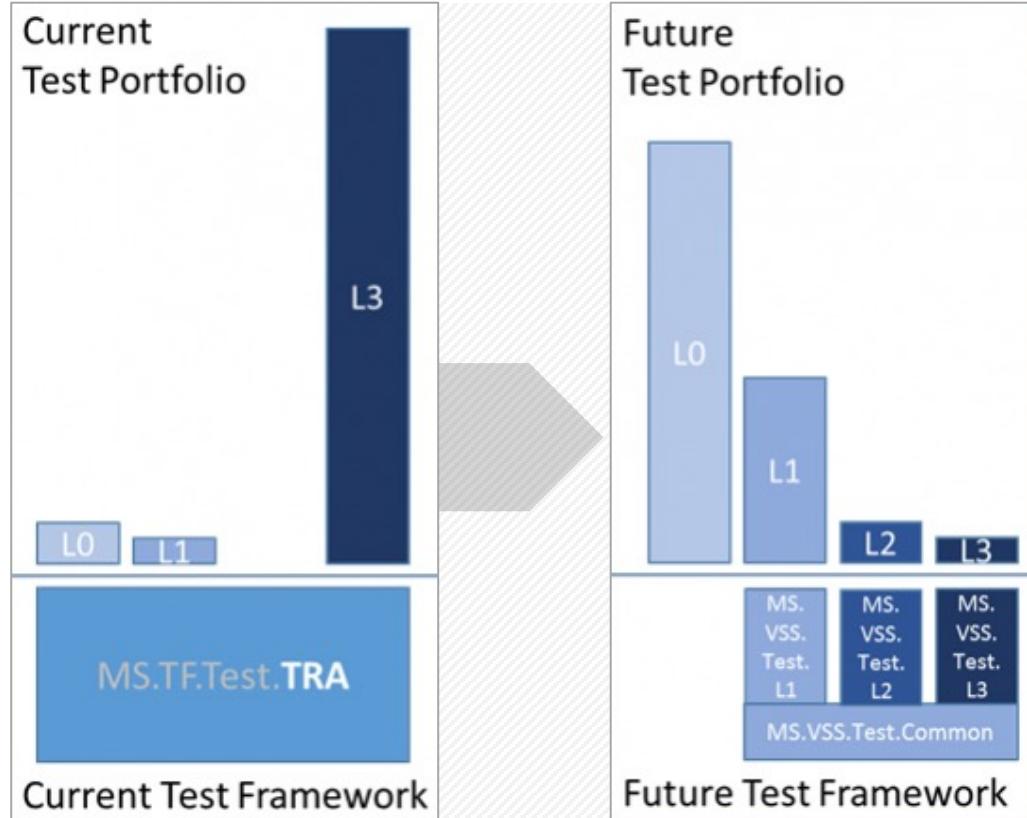
Quality approach designed for Cloud cadence

Build confidence through staging, exposure control & rich telemetry

Continue to meet a rigorous on-premises quality bar

Same definition of done must work for all delivery streams

Test at the lowest possible level



Principles

- Tests should be written at the lowest level possible
- Write once, run anywhere - includes production system
- Product is designed for testability
- Test code is product code, only reliable tests survive
- Testing infrastructure is a shared service

Our learnings as a SaaS provider



Talk to customers

Visual Studio

New and returning users may [sign in](#)

Welcome to the Visual Studio UserVoice site. Let us know what you would like to see in future versions of the Visual Studio suite of products. This site is for suggestions and ideas. If you need to file a bug, visit the Visual Studio Connect site: <https://connect.microsoft.com/visualstudio>.

To review the current UserVoice statuses and their definitions, please review our "[What Does the Status of My Feedback Mean?](#)" article.

We would also like to invite you to check out the [Announcements](#) section we have added to this site, where we will be posting special opportunities for you to participate in.

We look forward to hearing from you!

The Visual Studio Team
[Terms of Service](#) and [Privacy Policy](#)

I suggest you ...

Enter your idea

Hot Top New Team Foundation Server & Vis... Status My feedback

3,590 votes

Customize Process Template on Visual Studio Online
In Visual Studio Online, allow the ability to change the current process template a created project is using as well as the ability to upload your own process templates to create new projects with.

171 comments Team Foundation Server & Visual Studio Online Flag idea as inappropriate...

STARTED Visual Studio Team (Product Team, Microsoft) responded
I just published a blog post laying out the first pieces of our vision for VSO process customization. Have a look here: <http://blogs.msdn.com/b/visualstudioalm/archive/2015/05/05/visual-studio-online-process-customization.aspx>

Jan	Feb	Champ	Company	Users (30d)	(30d)	Builds (30d)	(30d)	Work items	Changesets	Test users	(30d)
2	1		Madhuri	121	2,623	1,050	1,635	26			
23	2		Tom	89	5,518	785	2,609	35			
5	3		Andrea	92	3,761	511	3,328	32			
3	4		Chris	111	1,090	1,975	2,452	42			
1	5		Ed	207	-	323	1,151	49			
4	6		Doug	109	1,609	-	1,423	3			
15	7		Will	79	1,426	174	1,552	41			
22	8		Vibhor	104	503	746	1,292	27			
8	9			92	969	22	1,426	58			
10	10			16	479	1,223	3,045	1			
37	11			12	590	597	3,012	0			
12	12			24	75	785	3,102	3			
13	13		Andrew	100	1,741	124	343	24			
20	14		Jon	33	664	1,039	433	24			
24	15		Andrea	74	530	480	3,128	10			
32	16		Jeff	111	1,000	243	652	32			
39	18			35	491	490	5,645	13			
99	19		Ewald	74	1,446	-	642	10			
62	20		Adam	127	563	-	-	2			
55	21		Mario	105	1,338	206	982	16			
200	22		Clemri	50	1,287	472	1,298	14			
73	23		Sarang	88	988	69	736	7			
126	26			32	703	1,037	656	18			
7	32		Manoj	69	1,028	312	1,442	37			
100	89		Tom	70	370	217	483	12			
25	90		Munil	31	684	446	592	18			
30	95		Manoj	69	595	-	567	40			
200	121		Harish	33	158	326	1,049	10			
200	200		Lori	14	90	34	121	5			
200	200		Federico	39	390	63	138	5			
81	200		Aaron	15	171	71	597	4			

Top Customers

UserVoice

Matching SLA to user experience

Phase 1: Outside-in synthetic tests

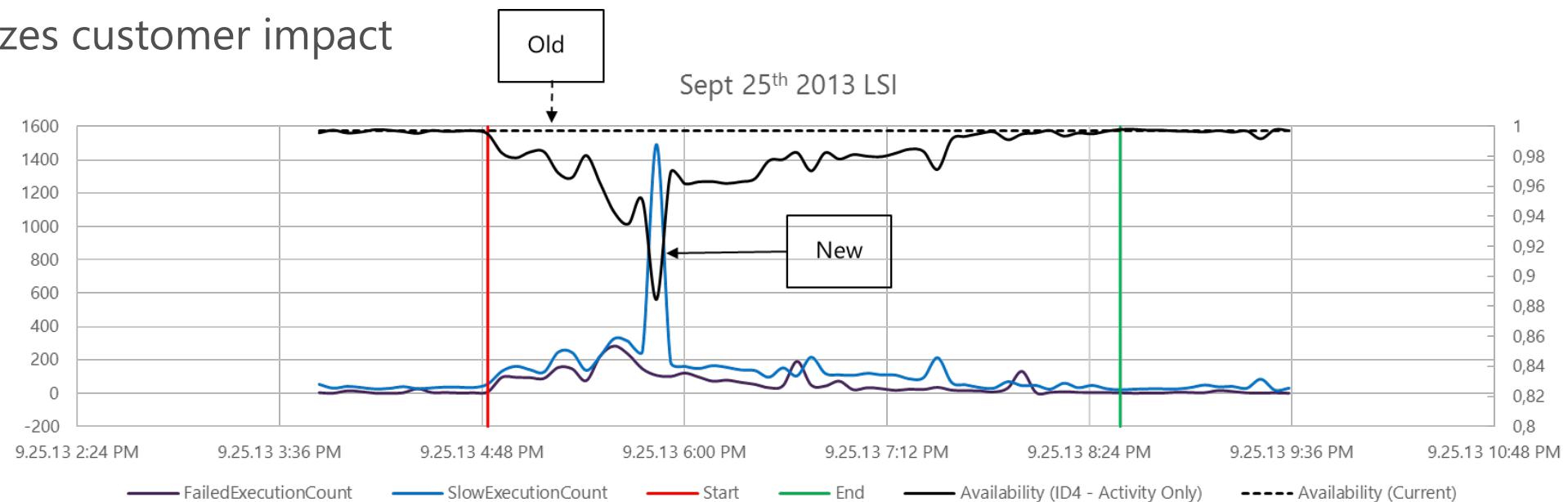
Experience: Coverage too narrow as service footprint grows

Phase 2: Command health

Experience: Loses sensitivity as command volumes grow

Phase 3: Failed or slow user minutes per account

Experience: Empathizes customer impact



Focus on the outliers ("Embrace the Red")

We measure availability by account ...

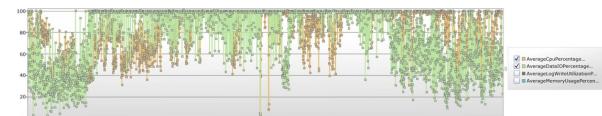
... and proactively reach out to customers with low availability

From: Ladislau Szomoru
Sent: Thursday, May 21, 2015 2:42 AM
To: Madhu Kavikondala; Harish Thekethil; Chandru Ramakrishnan; Ed Glas; Buck Hodges; Munil Shah
Subject: OneDrive.visualStudio.com

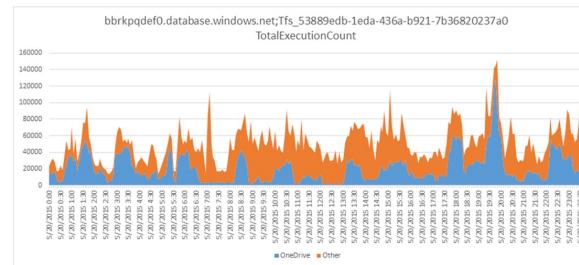
Hello All,

I am currently working on a stateful AIMS rule to fill a monitoring gap that we have around SQL Azure resource stats. I have already deployed the rule for testing purposes (alerts are routed to me only) and I am currently doing the analysis of the generated alerts in order to tweak the settings of the rule, as well as flush out any product bugs before I start routing the alert to VSOLS.

As part of this exercise I came across database bbrkpqdef0.database.windows.net;Tfs_53889edb-1eda-436a-b921-7b36820237a0 which is running very hot in CPU. Taking a closer look at the database we noticed that OneDrive.visualstudio.com is hosted in this database. We have already filed a product bug (323883) as prc_DeleteUnusedFiles that was running for hours on this account causing 100% CPU, but even after we have disabled the cleanup job, the database still [spikes up to 100% CPU](#).



Looking at the telemetry from 2015-05-20, I see that the OneDrive account represents 40% of the total commands in the problematic partition database. Below I have included a stacked area chart that breaks down the total commands between OneDrive and all the other accounts hosted in the partition database:



Based on the telemetry, I would like to do the following:

1. The problematic partition database is currently a P2, and I would like to ask for your permission to increase it to P3 in order to provide immediate relief to the other customers hosted in the same partition database. After the database is increased to P3, I will continue to look at the telemetry and peel the onion to make sure that any product bugs are being addressed.
2. Given the heavy usage of the OneDrive account, I believe that we should kick off the conversation with ASG about the long term plan of this account. If the account is here to stay, I would like to suggest that we move it to TFS SU2 in a single-tenant partition database hosted on the data tier that was purchased to ASG. Thoughts?

Let me know if you have any questions or concern regarding the telemetry. Thanks!

Ladislau

Found one of the top customers with low availability. Proactively reached out and resolved their issue.

Account level Availability Trend



We took inspiration from ...

Open Source Software (OSS) ...

OSS Workflows are all about sharing, reuse & collaboration

Componentization for sharing & collaboration are self re-enforcing

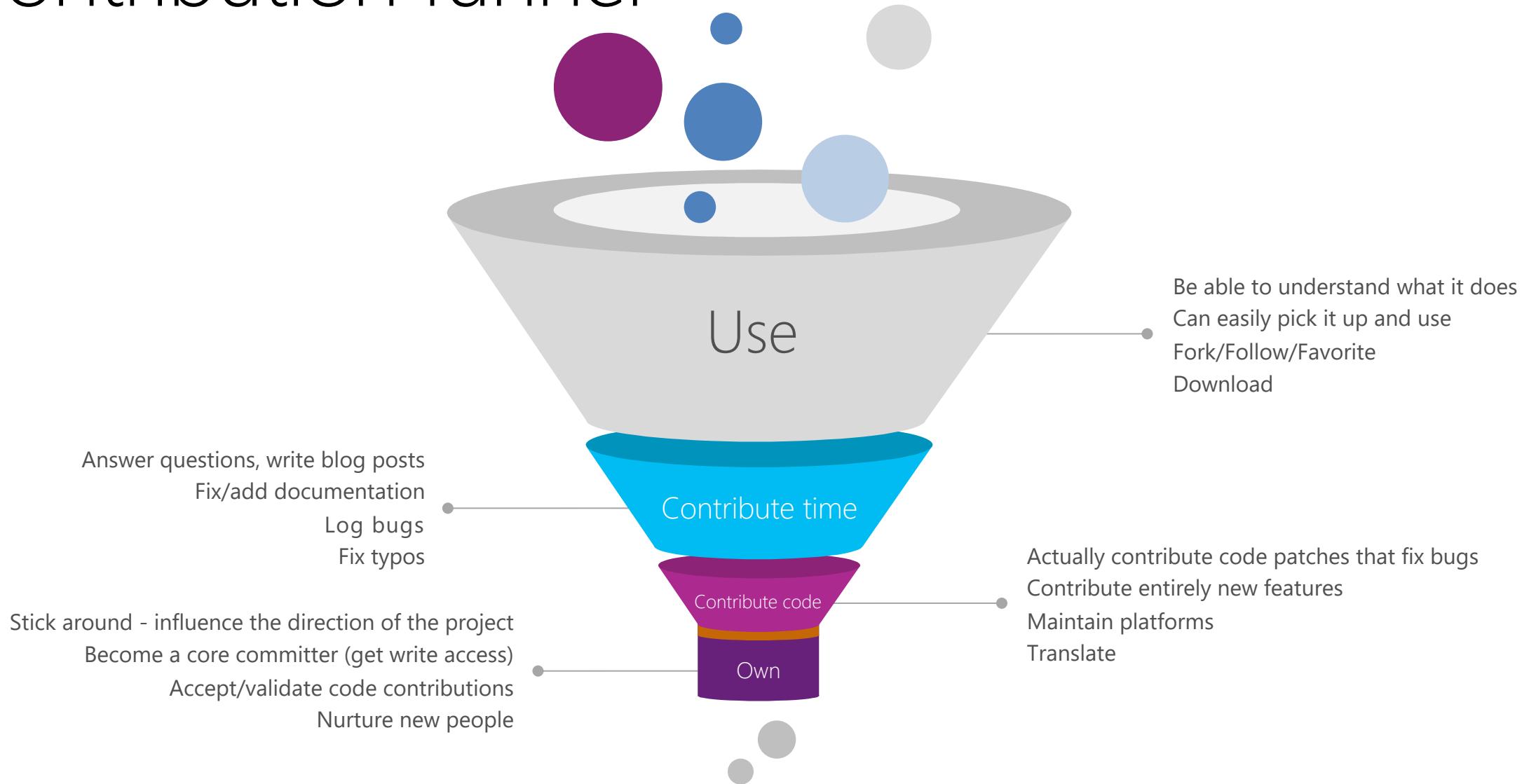
... and large-scale services ...

Loosely coupled, collaborating services that ship independently

... with Enterprise rigor

Enterprise code governance and requirements for products supported over a long period

Contribution funnel



Our learnings as a SaaS provider



Build-Measure-Learn

Hypothesis



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

Experiment



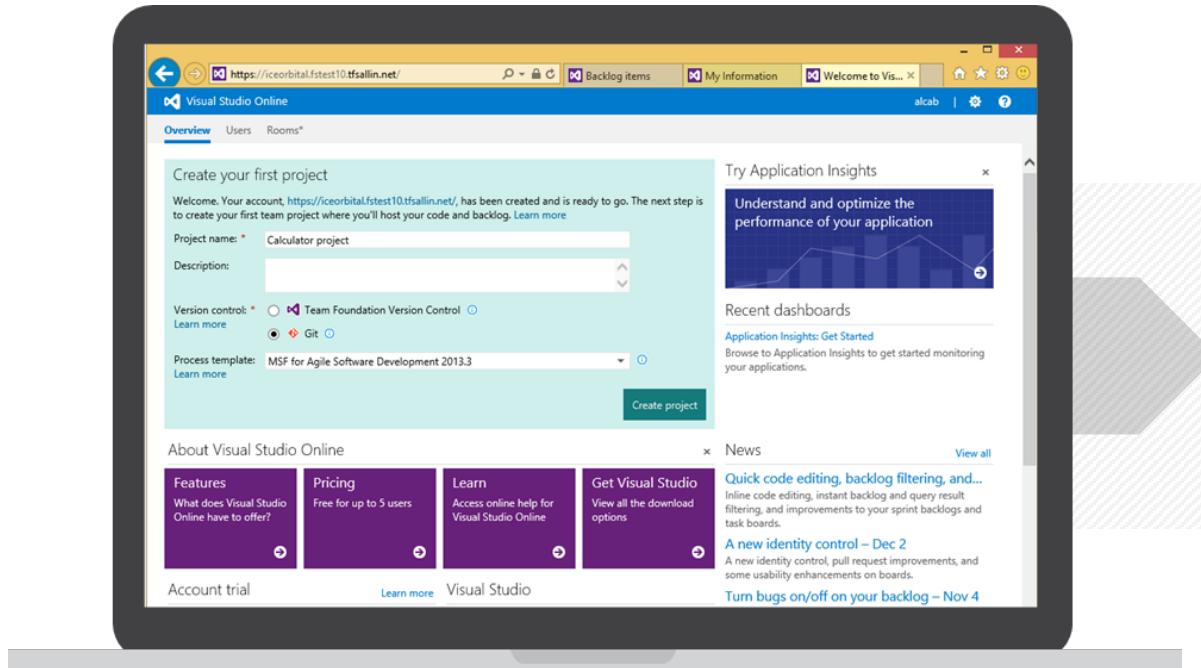
To prove or disprove the above, the team will conduct the following experiment(s): ...

Learning



The above experiment(s) prove(s) the hypothesis by impacting the following metric(s): ...

Before ...

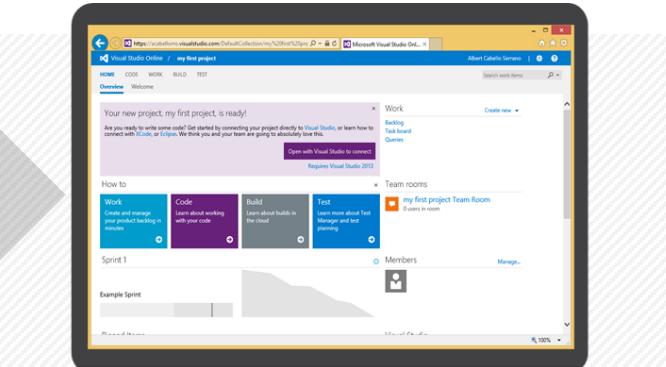
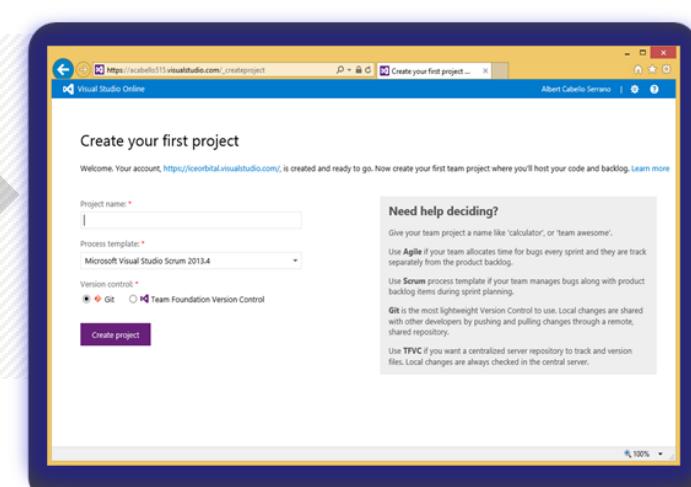
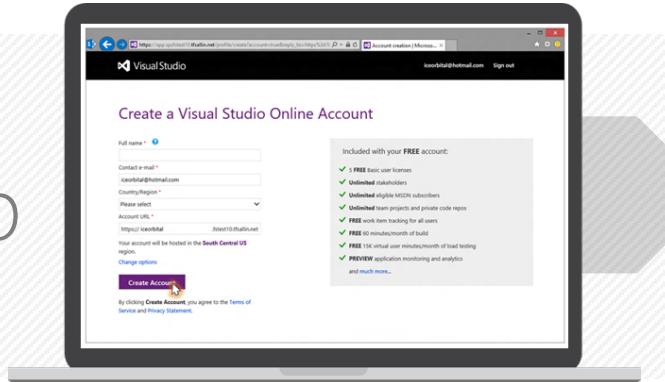


Previous project creation experience

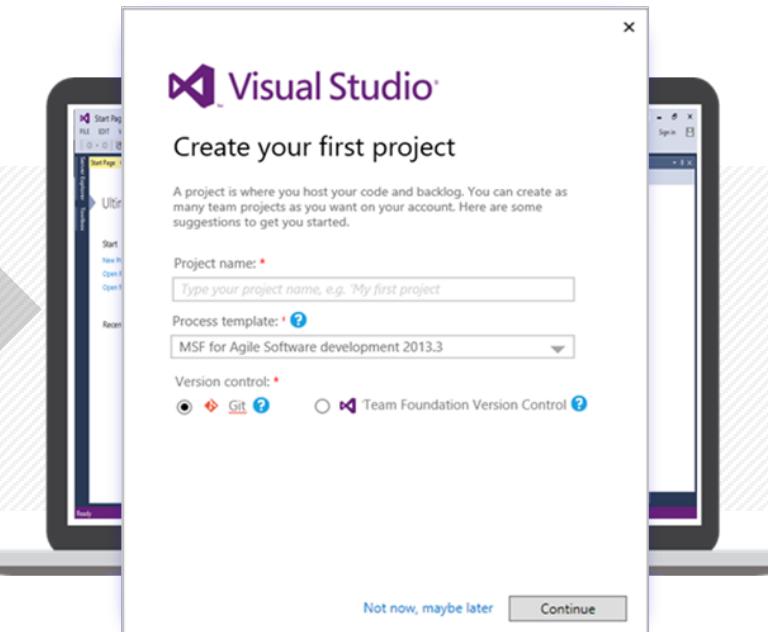
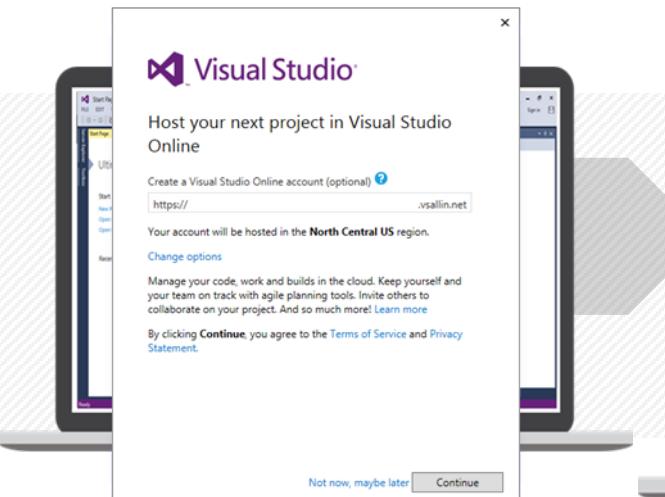
Too many actions and distractions can prevent customers from taking the next natural step of creating a new project

After ...

Web



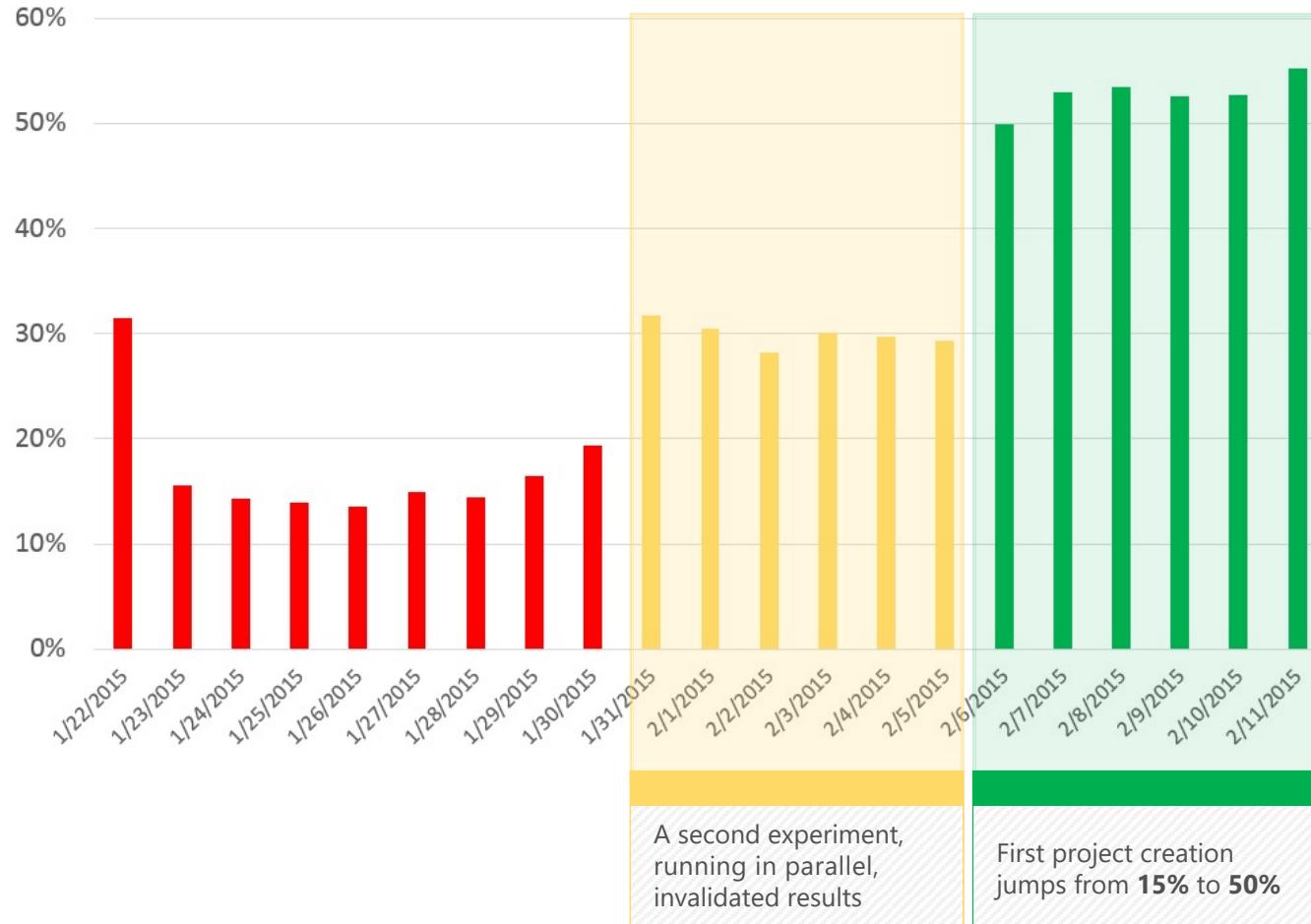
IDE



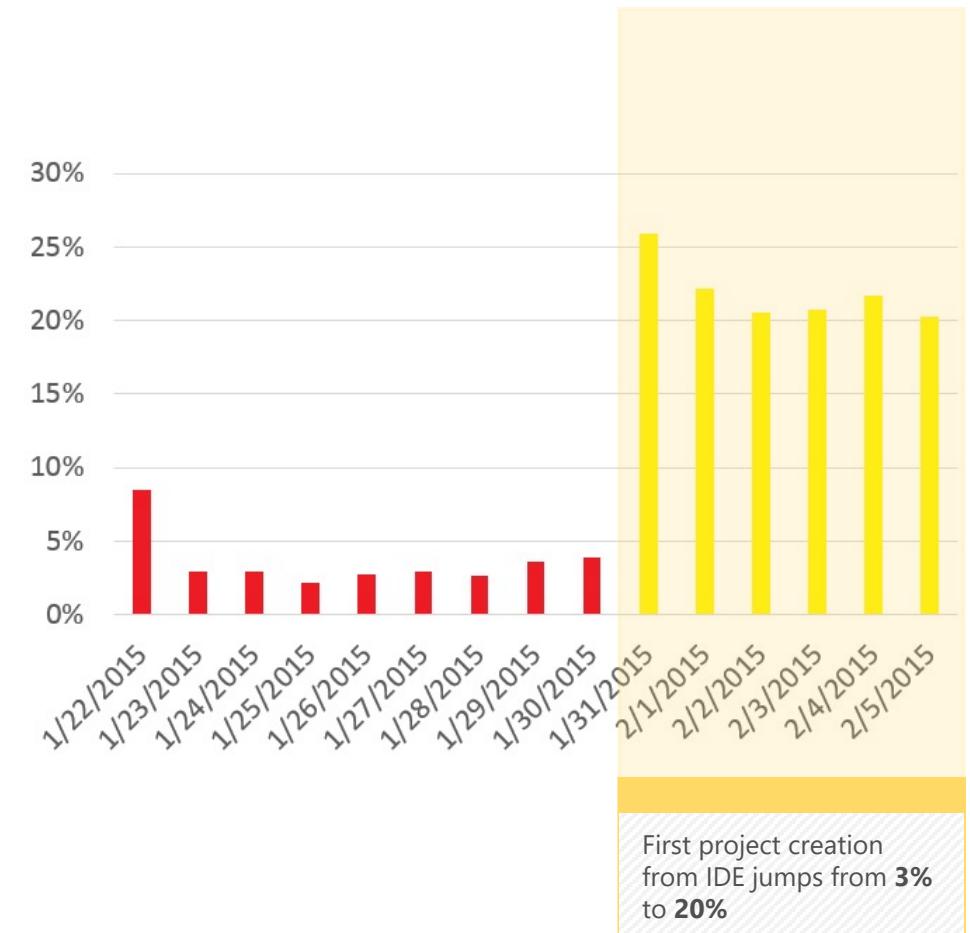
Focused project creation experience – integrated into the IDE flow

Validated learning

% of accounts creating projects the day of account creation
January 22 to February 11 (all sources)



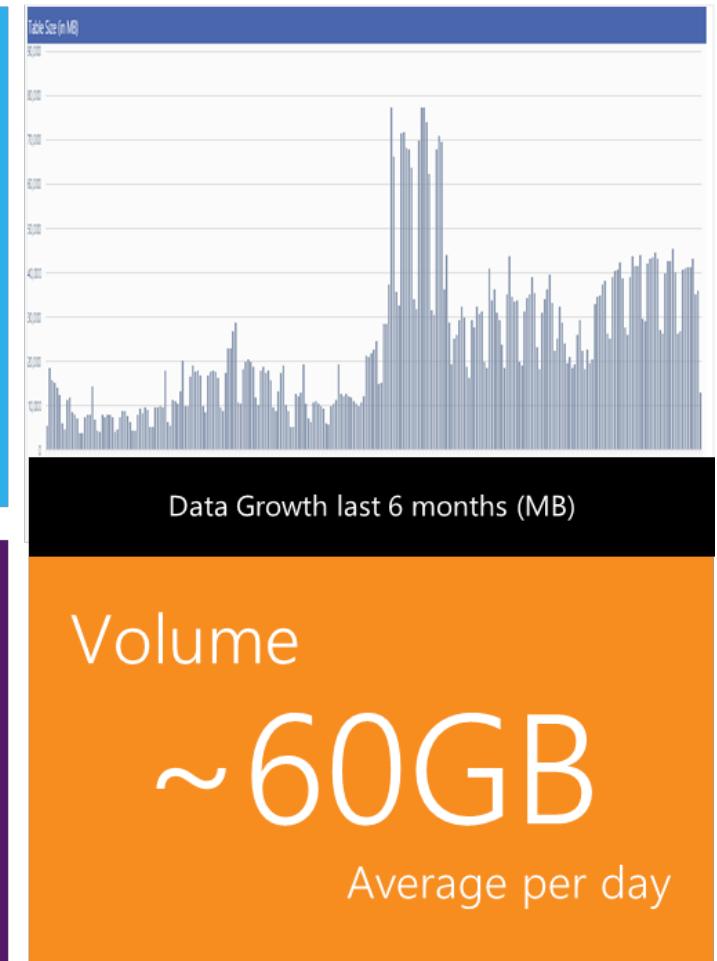
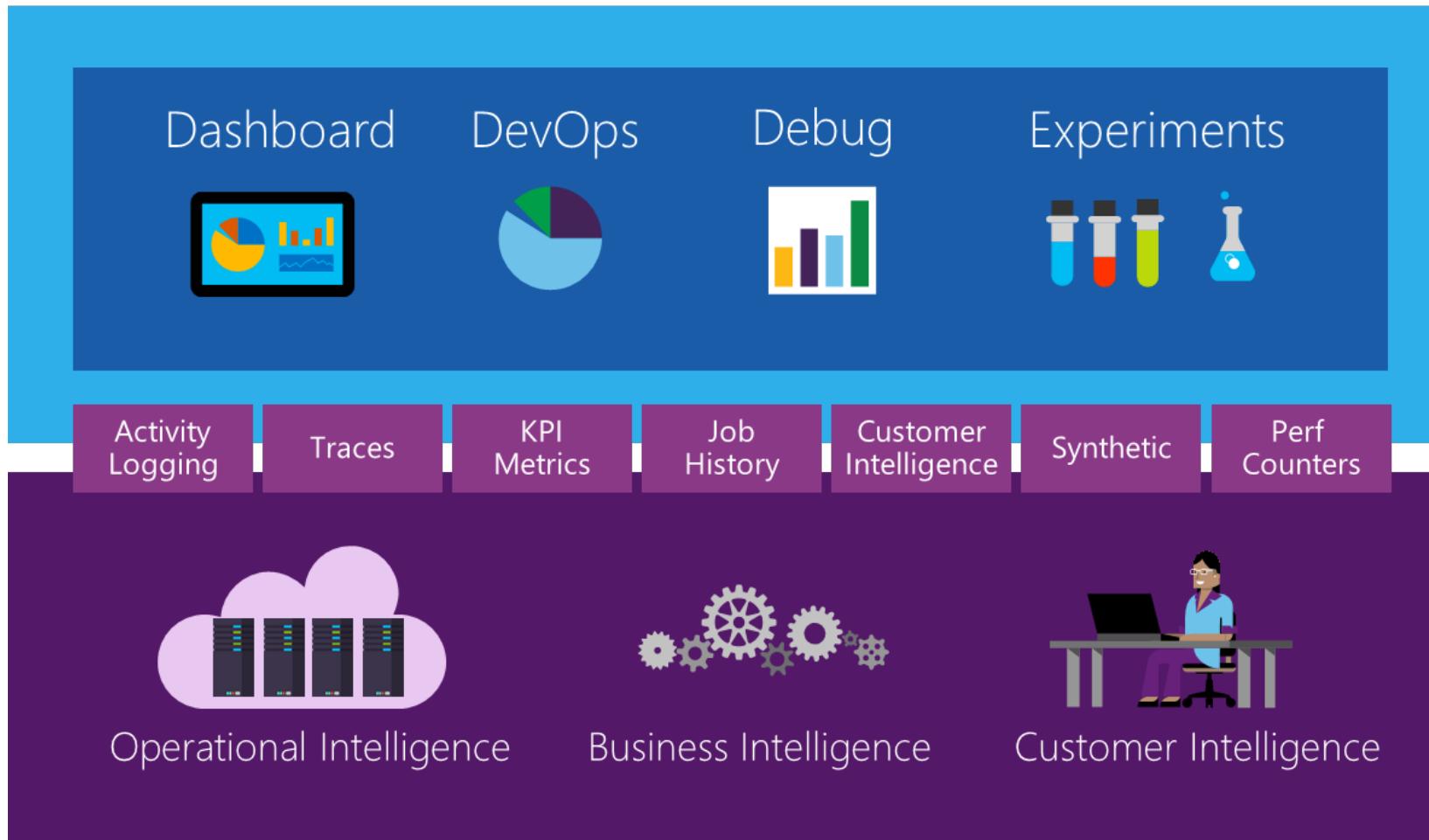
% of accounts creating projects the day of account creation
January 22 to February 5 (IDE only)



Our learnings as a SaaS provider

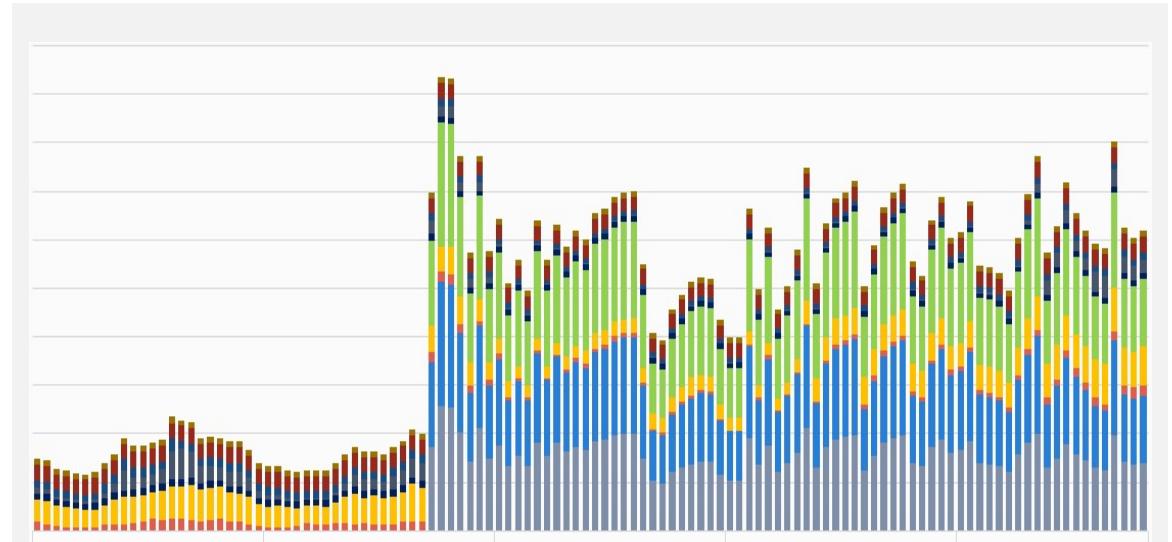


We collect everything!



Traces

Traces states in the code, error conditions
Lots of knobs (user, component, statement), all errors on by default
Great for debugging live site incidents
Up to 150gb per day



KPI metrics

Aggregated metrics that are “cooked” in real-time or in the back-end

Used to calculate availability and generate alerts

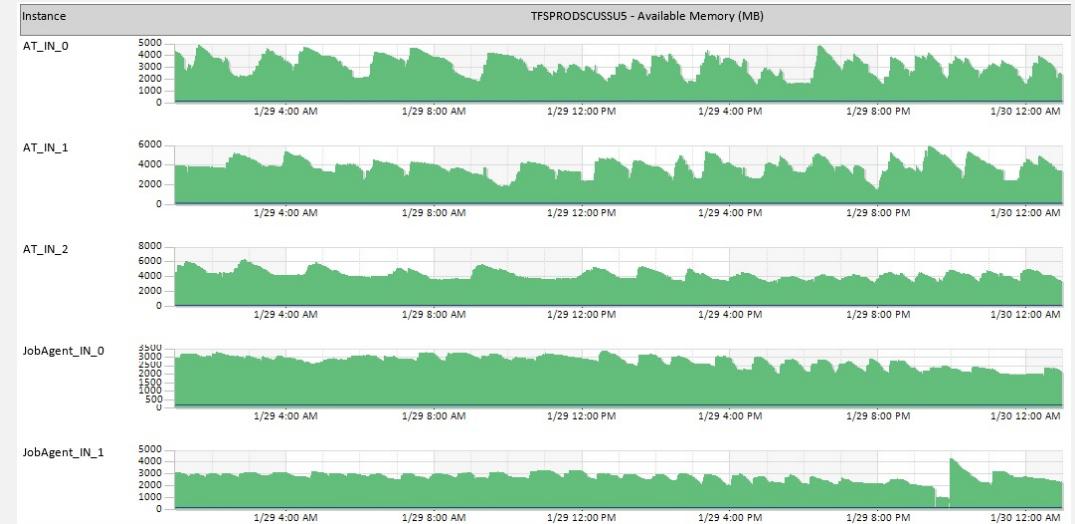


Perf counters

Capture performance of the platform and Azure infrastructure

Use for debugging and measuring performance

~50M events per day



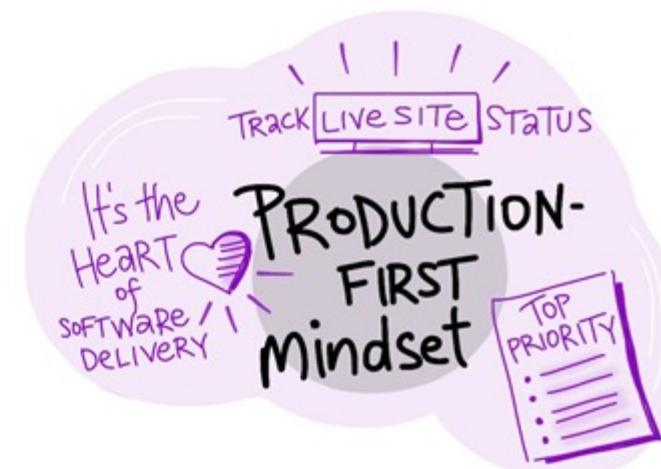
Ping mesh

Capture a baseline of network performance and health

Helps to narrow down network related problems quickly – problems external to the system are harder to identify and take longer to escalate



Our learnings as a SaaS provider



Live-site culture principles

Site status is always priority 1

Global response team

On-call DRI (Designated Responsible Individual) by area

Service Delivery team in 5 time zones for 24x7

Weekly live site review

Monthly service review

Communicate in real time

Live-site issues become product backlog items

Fix at *root cause*

Automate solutions

Live Site Issues (LSIs)

Visual Studio Team Foundation Server 2013 / DevDiv

Sam Guckenheimer | ?

HOME CODE WORK BUILD TEST RELEASE

Search work items

Backlogs Queries

New | Assigned to me Unsaved work items

Recent work items

Live Site Inciden...

My favorites Drag queries here...

Team favorites SD Bugs - SPS TFS QA Remain...

My Queries Shared Queries

Live Site Incident 1457975: Pop node certificate "Outsideln.RegisterNode" expired

Tags Add...

Title: Pop node certificate "Outsideln.RegisterNode" expired ID #: 1457975

Area Path: DevDiv\Application Insights\Live Site Experience

Iteration Path: DevDiv\VS Cloud Services

INCIDENT DETAILS

Created Date:	1/10/2014 1:08:25 PM
Service Category:	Applnights
Impacted Service:	GSM
Incident State:	5 – Closed
Severity:	1 – Severe Impact(>1% to 80% customer impa
Assigned To:	David Goddard
Detection Method:	Automated
Detection Source:	Applnights
Environment:	Production
Datacenter:	
Instance Name:	

INCIDENT TIMELINE

Incident Start Time:	1/10/2014 1:06:25 PM
Detected:	1/10/2014 1:09:00 PM
Triaged - VSOLS:	1/10/2014 1:11:00 PM
Escalated - SE:	1/10/2014 1:13:00 PM
Acknowledged - SE:	1/10/2014 1:13:00 PM
Partner Engaged:	
Engaged - SE:	1/10/2014 1:13:00 PM
Incident End Time:	1/10/2014 2:54:00 PM
External Communication:	1/10/2014 1:36:00 PM
Internal Communication:	1/10/2014 1:34:00 PM
SE-Effort(hrs):	
SD-Team Count:	

LIVE SITE REVIEW

Customer Impact:	Yes
Live Site Review?:	Yes
LSR Owner:	David Goddard
LSR Owning Role:	2 - Ops
Partner Id:	
Repeat issue?:	No
Resolution type:	Resolved By SE
KB ID#:	1396574
Error Category:	Application
Error Source:	Maintenance
Error Source SubCategory:	
Alert count:	

SUMMARY IMPACT MITIGATION RCA PROBLEM MANAGEMENT (2) ATTACHMENTS (1) NOTES HISTORY MISC VSOLS

MITIGATION

B / U

Outage Window

Healthy Status

1/10/2014 1:38 AM 01/10/2014 11:50 AM 01/10/2014 12:02 PM 01/10/2014 12:14 PM 01/10/2014 12:25 PM 01/10/2014 12:37 PM 01/10/2014 12:49 PM 01/10/2014 1:01 PM 01/10/2014 1:13 PM 01/10/2014 1:25 PM 01/10/2014 1:37 PM 01/10/2014 1:48 PM 01/10/2014 2:00 PM 01/10/2014 2:12 PM 01/10/2014 2:24 PM

Auto-dumper

Root Cause

Get diagnostics the first time an issue occurs

Secure

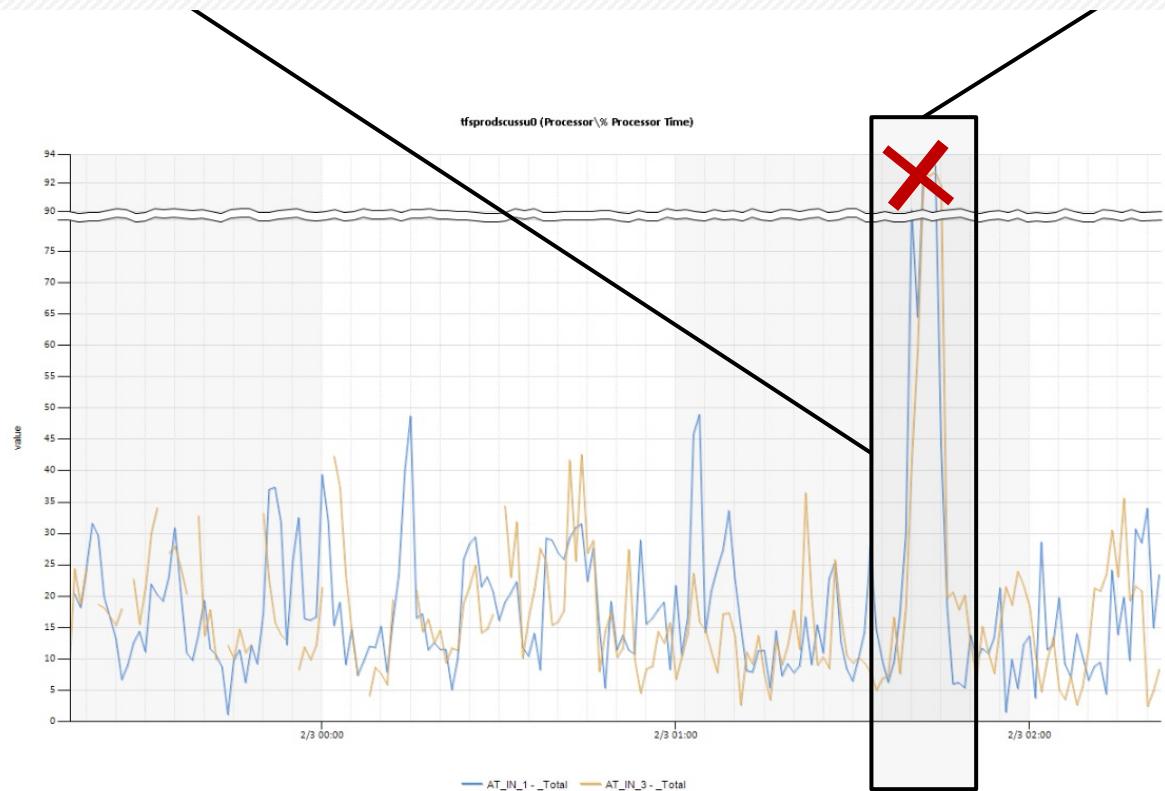
Avoid using RDP which reduces security risk

Fast

Avoid manual repro and data collection

Transient CPU spike. Hard to repro and get diagnostic tools enabled on time (dump, perfview)

Automated trigger can start the diagnostic process automatically and collect data on-demand



Automating communication

Save time

Automatically post to blog, email and twitter

Consistency

Leverage pre-approved templates

Approved templates make it easy to create an incident notification quickly. During the crisis it also helps the team remember what is required.

AI Service Blog

Title:

Data Loss issue with Web Test Data - 2/13 - Resolved



Final Update: Saturday, 2/14/2015 03:54 UTC

We've confirmed that all systems are back to normal with no customer impact as of 2/13/2015 20:48 UTC. Our logs show the incident started on 2/13/2015 17:30 UTC and that during the 3 hours that it took to resolve the issue customers experienced 25% - 50% data loss for application dependency data.

- [Root Cause](#): The failure was due to noisy neighbor, which impacted our ability obtain a blob lease in storage.
- [Chance of Reoccurrence](#): Low
- [Lessons Learned](#): We are investigating additional service improvements and optimizations to avoid this issue in the future.
- [Incident Timeline](#): 3 Hours & 18 minutes - 2/13/2015 17:40 UTC through 2/13/2015 20:48 UTC

We understand that customers rely on Application Insights as a critical service and apologize for any impact this incident caused.

-Application Insights Service Delivery Team

Update: Saturday, 2/14/2015 02:31 UTC

While restoring the original configuration, two instances became non-responsive. DevOps is investigating those instances. There is currently no impact to customers.

- [Next Update](#): 2/14/2015 06:00 UTC

-Application Insights Service Delivery Team

Twitter Post

Incident

Investigating (Initial Note)

Investigating (No RCA)

Mitigating (Known RCA)

Resolved

Root Cause

PST 02/14/2015 09:14

UTC 02/14/2015 17:14

February 2015

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

Time 00:00

The screenshot shows a software interface for managing incident communications. On the left, there's a rich-text editor window with a toolbar. The main content area displays a blog post about a data loss issue, with several sections of text and a bulleted list of details. To the right of the editor is a sidebar containing various incident status options like 'Investigating' and 'Resolved', along with time zone settings (PST and UTC) and a calendar for February 2015. A prominent black arrow originates from the text block at the bottom left and points directly at the sidebar.

Service status visible

Profile summary X



VSONLINE
@vsonline

Followed by Buck Hodges, Ed Blankenship, Aaron Bjork and Visual Studio.

VSONLINE @vsonline · Jan 23
Update to #VSONLINE planned for Monday, 26 Jan through Thursday, 29 Jan.
See Service Blog at blogs.msdn.com/b/vsoservice/ for details.
[Details](#)

VSONLINE @vsonline · Jan 15
@vsonline The underlying Azure issue has been resolved and we are processing the backlog of requests. We apologize for the inconvenience
[Details](#)

VSONLINE @vsonline · Jan 15
@jchandra Sorry to hear that. We're working with our partners in Azure to bring things back online as soon as possible.
[Details](#)

 **Visual Studio Online is up and running**

Everything is looking good
For details and history, check out the [Visual Studio Service Blog](#).

Service Blog - Visual Studio Online



Sort by: [Most Recent](#) | [Most Views](#) | [Most Comments](#)

Issues with Application Insights Configuration service 2/15-Mitigated

1 day ago by [Visual Studio Online Team](#)

Final Update: Wednesday 2/15/2015 21:30 UTC The networking issue has been resolved as of 2/15/2015 20:16 UTC. Customers will be able to create and view web tests without any issue. Thank you for your patience. -Application Insights Service...

Maintenance Monday, 16 Feb - Thursday, 19 Feb - Scheduled

3 days ago by [Visual Studio Online Team](#)

We will be shipping an update to Visual Studio Online. We are planning on having maintenance windows from Monday, 16 Feb through Thursday, 19 Feb. There should be no impact to the service during this update. Thanks, Erin Dormier

Data Loss issue with Application Insights Web Test Data - 2/13 - Resolved

3 days ago by [Visual Studio Online Team](#)

Final Update : Saturday, 2/14/2015 03:54 UTC We've confirmed that all systems are back to normal with no customer impact as of 2/13/2015 20:48 UTC. Our logs show the incident started on 2/13/2015 17:30 UTC and that during the 3 hours and 18 minutes...

Data Loss issue with Application Insights - 2/11 - Resolved

4 days ago by [Visual Studio Online Team](#)



Final Update : Wednesday, 2/11/2015 22:41 UTC We've confirmed that all systems are back to normal as of 2/11, 22:28 UTC. Our logs show the incident started on 2/11, 21:10 UTC and that during the 78 minutes that it took to resolve the issue...

Issues with Application Insights Web Test Data from Moscow Datacenter - 2/10 -

RCA (Root Cause Analysis) transparency

A Rough Patch

Brian Harry MS 25 Nov 2013 3:06 PM 10

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

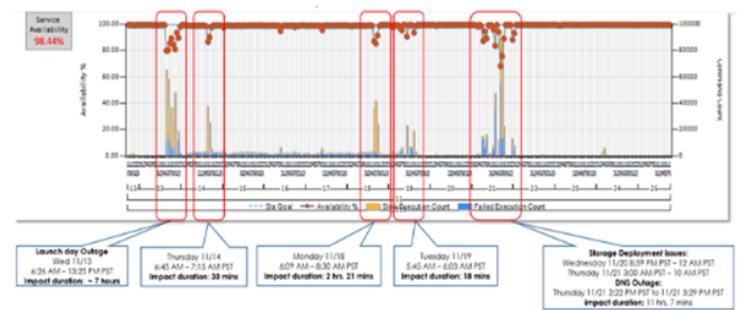
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you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

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

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

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

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



Explanation of July 18th outage

Brian Harry MS 31 Jul 2014 5:58 AM 6

RATE THIS

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

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

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

What happened?

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

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

The slow Azure database caused calls to the Shared 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 and 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 SBS/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.

Retrospective on the Aug 14th VS Online outage

Brian Harry MS 22 Aug 2014 11:10 AM 20

RATE THIS

We had a pretty serious outage last Thursday all told it was a little over 5 hours. The symptoms were that performance was so bad that the service was basically unavailable for most people (though there was some intermittent access as various mitigation steps were taken). It started around 14:00 UTC and ended a little before 19:30 UTC. This duration and severity makes this one of the worst incidents we've ever had on VS Online.

We feel terrible about it and continue to be committed to doing everything we can to prevent outages. I'm sorry for the problems it caused. The team worked tirelessly from Thursday through Sunday both to address the immediate health issues and to fix underlying bugs that might cause recurrences.

As you might imagine, for the past week, we've been hard at work trying to understand what happened and what changes we have to make to prevent such things in the future. It is often very difficult to find proof of the exact trigger for outages but you can learn a ton by studying them closely.

On an outage like this, there's a set of questions I always ask, and they include:

What happened?

What happened was that one of the core SPS (Shared Platform Services) databases became overwhelmed with database updates and started queuing up so badly that it effectively blocked callers. Since SPS is part of the authentication and licensing process, we can't just completely ignore it – though I would suggest that if it became very sluggish, it wouldn't be the end of the world if we bypassed some licensing checks to keep the service responsive.

What was the trigger? What made it happen today vs yesterday or any other day?

Though we've worked hard on this question, we don't have any definitive answer (we're still pursuing it though). We know that before the incident, some configuration changes were made that caused a significant increase in traffic between our "TFS" service and our "SPS" (Shared Platform Service). That traffic involved additional license validation checks that had been improperly disabled. We also know that, at about the same time, we saw a spike in latencies and failed deliveries of Service Bus messages. We believe that one or both of these were key triggers but we are missing some logging on SPS database access to be able to be 100% certain. Hopefully, in the next few days, we'll know more conclusively.

What was the "root cause"?

This is different than the trigger in the sense that the trigger is often a condition that caused some cascading effect. The root cause is more about understanding why the effect cascaded and why it took the system down. It turns out that, I believe, the root cause was that we had accumulated a series of bugs that were causing extra SPS database work to be done and that the system was inherently unstable – from a performance perspective. It just took some poke at the system – in the form of extra identity or licensing churn to cause a ripple effect on these bugs. Most, but not all, of them were introduced in the last few

Precise alerting is key to fast detection ...

... but poor alerting makes DevOps unhappy

Redundant alerts for same the issue

Needed to set right thresholds and tune often

Stateless alerts contributed to further noise

We set alerting goals

Every alert must be actionable and represent a real issue with the system

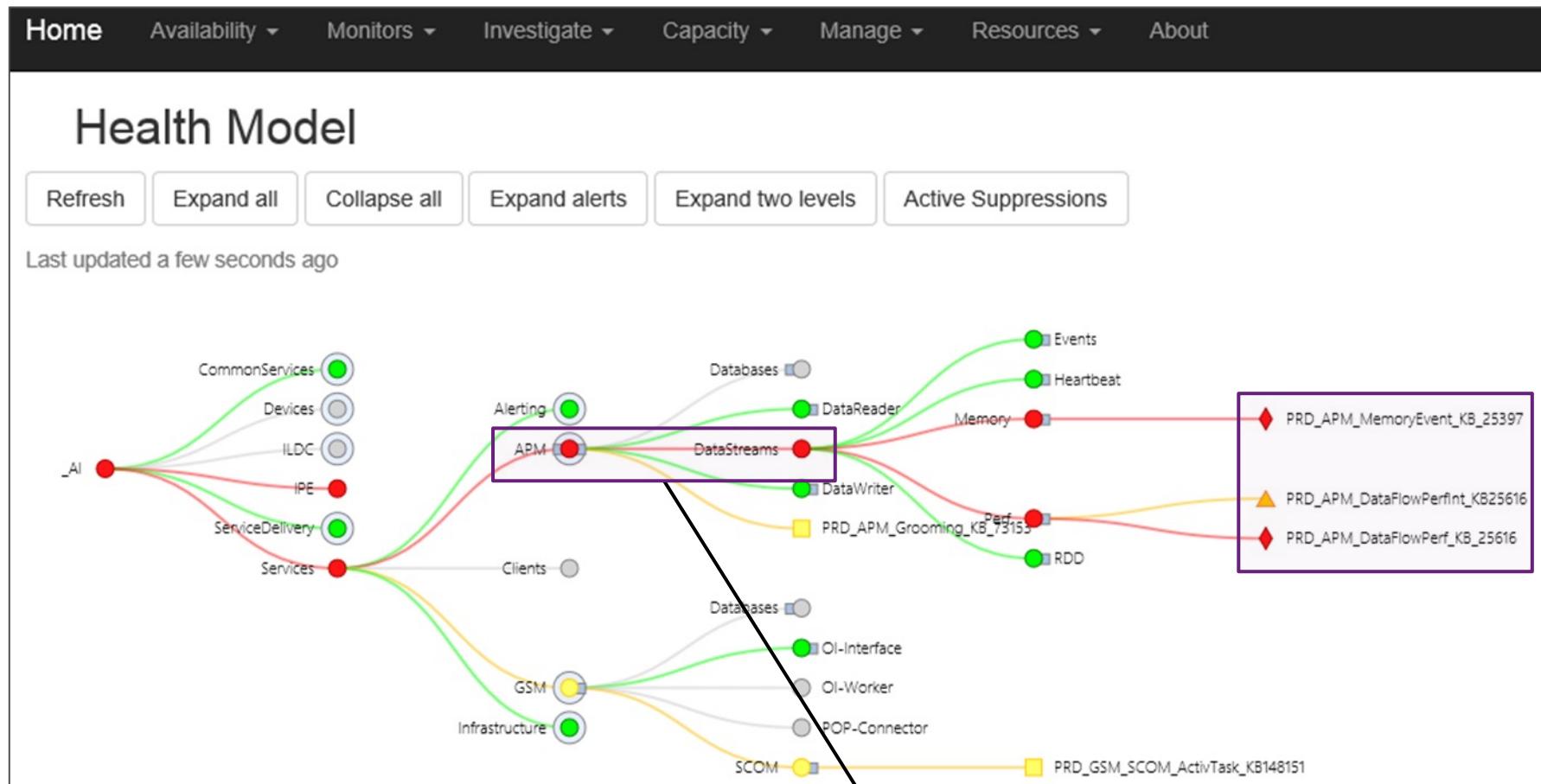
Alerts should create a sense of urgency – false alerts dilute that

Solution

Consolidate alerts so that only actionable alerts are sent to team

Autoroute according a health model based on suspect route cause

Health model in action



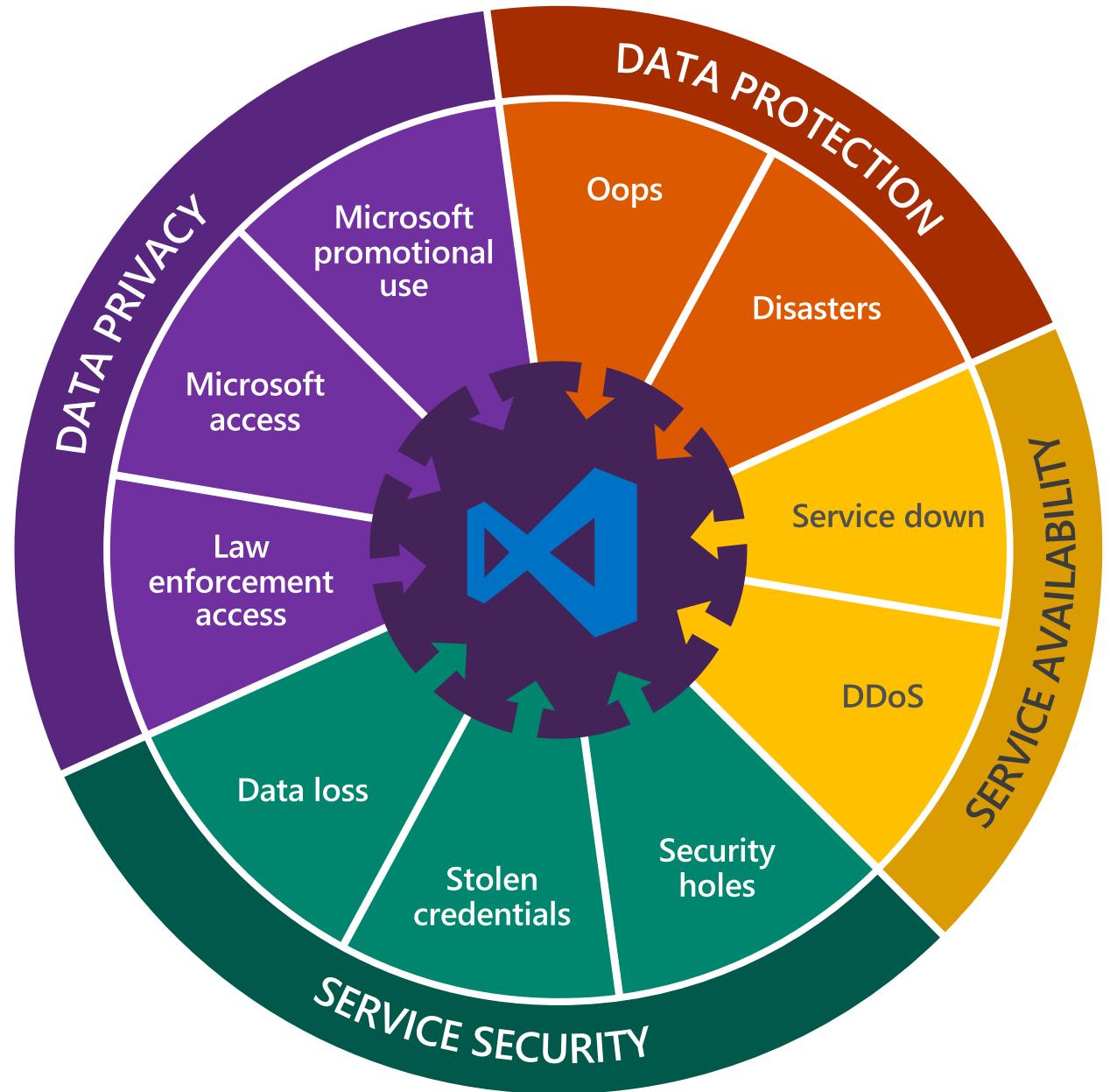
Found 3 errors for memory and performance
All 3 errors are related to the same code defect

APM component mapped to feature team
Auto-dialer engaged Global DRI

Eliminated alert noise:
~928 alerts per week to ~22
Reduced DRI escalations by ~56%

Our security mindset is
assume breached, not
just prevent breach!

Comprehensive security program
designed with this mindset



Secrets management

Secrets

Certificates
Storage keys
Passwords
Service accounts

Challenges

Expiry
Expiry date associates with all secrets
Leak
Accidental exposure of secrets
Theft
Hacking attempts and security breaches

How we do it

Azure secret store
Encrypted store w/ self-service API
Encrypted config
Encrypt secrets in app config
Expiring object scanner
Scan and alert on secret expiry date
Rotate
Change secrets regularly
Audit
Alert on unauthorized access

1000+
Secrets in VSO



Bad day

Brian Harry MS 23 Feb 2013 4:29 AM 11



Clearly yesterday was a bad day. Team Foundation Service was mostly down for approximately 9 hours. The underlying issue was an expired SSL certificate in Windows Azure storage. We use HTTPS to access Windows Azure storage, where we store source code files, Git repos, work item attachments and more. The expired certificate prevented access to any of this information, making much of the TFService functionality unavailable.

We were watching the issue very closely, were on the support bridge continuously and were investigating options to mitigate the outage. Unfortunately we were not successful and had to wait until the underlying Azure issue was resolved. I have a new appreciation for the "fog of war" that happens so easily during a large scale crisis. We'll be sitting down early this week to go through the timeline hour by hour – what we knew, what we didn't know, what we tried, what else we could have tried, how we communicated with customers and everything else to learn everything we can from the experience.

Our learnings as a SaaS provider



Automating deployments

Deploy anytime

Online operation – service stays up 24x7x365

Fully automated through TFS Release Management

Multiple versions must coexist peacefully

Staging

Canary stamp first (SU0), then other instances

Detailed health analysis after each stamp

Exposure control using feature flags

Enable or disable features at runtime without code changes or redeployment

Progressive disclosure of features

Release Management

This is the preview version of Release Management service. Refer to [documentation](#) for more information.

Docker CD | Edit

Releases Overview State All

Release Definitions

All release definitions

Docker CD

Release	Title	Environments	Build	Branch	Date	Created By	Description
+ Release-19	Release-19	[Green]	334		19 hours ago	Sam Guckenheimer	Triggered by Docker CI 334.
+ Release-18	Release-18	[Green, Gray]	333		19 hours ago	Sam Guckenheimer	Triggered by Docker CI 333.
+ Release-17	Release-17	[Red]	332		20 hours ago	Sam Guckenheimer	Triggered by Docker CI 332.
+ Release-16	Release-16	[Green]	331		21 hours ago	Sam Guckenheimer	Triggered by Docker CI 331.
+ Release-15	Release-15	[Green, Red]	328		10/3/2015	Donovan Brown	
+ Release-14	Release-14	[Green, Red]	328		9/25/2015	Donovan Brown	Triggered by Docker CI 328.
+ Release-13	Release-13	[Green, Red]	326		9/24/2015	Donovan Brown	Triggered by Docker CI 326.
+ Release-12	Release-12	[Green, Red]	325		9/24/2015	Donovan Brown	Triggered by Docker CI 325.
+ Release-11	Release-11	[Green, Red]	324		9/24/2015	Donovan Brown	Triggered by Docker CI 324.
+ Release-10	Release-10	[Green, Red]	323		9/24/2015	Donovan Brown	Triggered by Docker CI 323.
+ Release-9	Release-9	[Green, Red]	322		9/24/2015	Donovan Brown	Triggered by Docker CI 322.
+ Release-7	Release-7	[Green, Red]	320		9/24/2015	Donovan Brown	Triggered by Docker CI 320.
+ Release-6	Release-6	[Green]	318		9/23/2015	Donovan Brown	Triggered by Docker CI 318.
+ Release-5	Release-5	[Green, Red]	317		9/23/2015	Donovan Brown	Triggered by Docker CI 317.
+ Release-4	Release-4	[Green, Red]	316		9/23/2015	Donovan Brown	Triggered by Docker CI 316.
+ Release-3	Release-3	[Green]	315		9/23/2015	Donovan Brown	Triggered by Docker CI 315.
+ Release-2	Release-2	[Green]	314		9/23/2015	Donovan Brown	Triggered by Docker CI 314.
+ Release-1	Release-1	[Green]	312		9/23/2015	Donovan Brown	Triggered by Docker CI 312.
✓ Release.20150921131033		[Green]	308		9/21/2015	Donovan Brown	Triggered by Docker CI 308.
✓ Release.20150921115656		[Green]	307		9/21/2015	Donovan Brown	Triggered by Docker CI 307.
✓ Release.20150920214657		[Green]	306		9/20/2015	Donovan Brown	Triggered by Docker CI 306.
✓ Release.20150920212044		[Green]	305		9/20/2015	Donovan Brown	Triggered by Docker CI 305.
✓ Release.20150918232955		[Green]	304		9/18/2015	Donovan Brown	Triggered by Docker CI 304.
✓ Release.20150918213851		[Green]	303		9/18/2015	Donovan Brown	Triggered by Docker CI 303.

Feature flags

All code is deployed, but feature flags control exposure

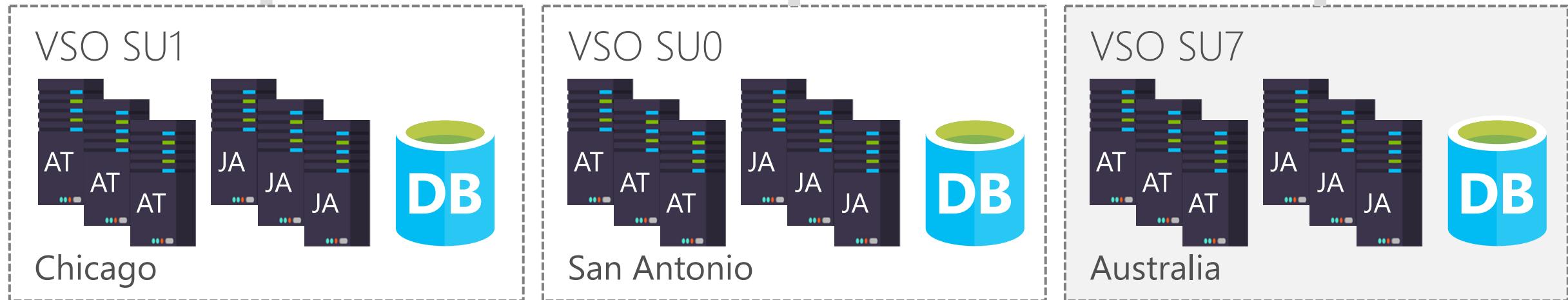
Flags provide runtime control down to individual user

Users can be added or removed with no redeployment

Enables dark launch

Mechanism for progressive experimentation & refinement

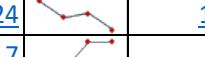
Multiple scale units enable canarying

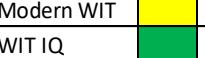
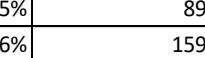
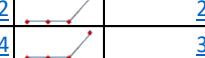
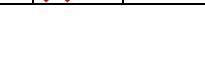
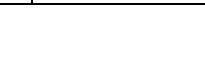
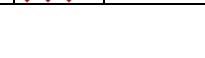


Our learnings as a SaaS provider



Metrics we track

Feature Team	Livesite Health and Debt					
	% of LSIs automatically detected	Max TTM (mins)	# LS prevention work items	LSPs rank <= 50 & P2 or higher LSI work > 21d	# of DTS over SLA	
Modern WIT	NA		NA		<u>24</u> 	<u>15</u> 
WIT IQ	NA		NA		<u>7</u> 	<u>3</u> 

Feature Team	Engineering Debt						
	# Bugs resolved/ engr in last 21 days	# Active bugs per engr	# P0 or P1 bugs > 21 days	Team Azure NAR % pass	Team Azure NAR time (mins)	Test Improvement	Security: # WI > 21 days
Modern WIT	11 	8 	8 	92.5%	89	2 	2 
WIT IQ	6 	3 	5 	94.6%	159	4 	3 

Live Site Health

Time to Detect
Time To Mitigate
Incident prevention items
Aging live site problems
Customer support metrics
(SLA, MPI, top drivers)

Velocity

Time to build
Time to self test
Time to deploy
Time to learn
(Telemetry pipe)

Engineering

Bug cap per engineer
Aging bugs in important categories
Pass rate & coverage

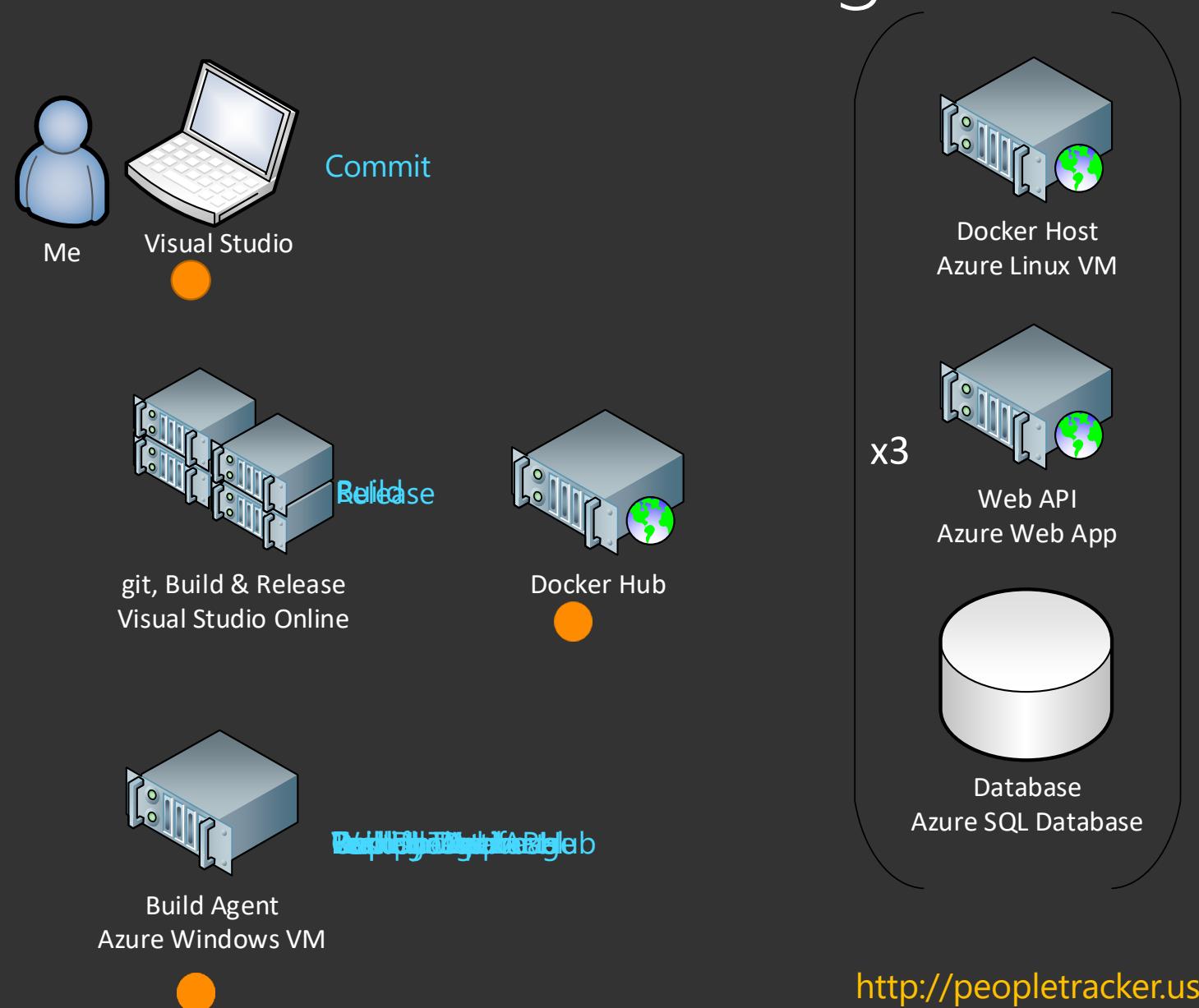
Usage

Acquisition
Engagement
Retention
Conversion
Churn

DEMO: Productizing Our Learnings

Automate Deployment with Release Management

Containers across Windows & Linux



There's no place
like production!





© 2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows 7, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. Microsoft makes no warranties, express or statutory, as to the information in this presentation.