Moving to One Engineering System

Sam Guckenheimer
Visual Studio Team Services
Microsoft

Microsoft

About Me

Sam Guckenheimer

Product Owner, Visual Studio Cloud Services

13 years Microsoft

30 years software industry

@SamGuckenheimer

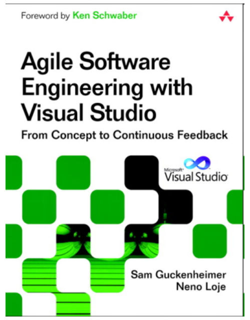
https://visualstudio.com/devops

Foreword by Ivar Jacobson, Ph.D.

Software Engineering with Microsoft Visual Studio Team System

Microsoft Visual Studio Team System

Sam Guckenheimer *with* Juan J. Perez



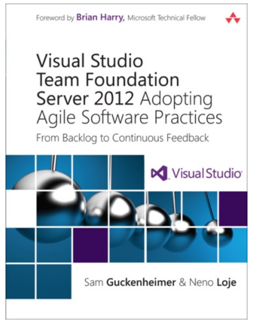
Foreword by Ken Schwaber

Agile Software Engineering with Visual Studio

From Concept to Continuous Feedback

Microsoft Visual Studio

Sam Guckenheimer
Neno Loje



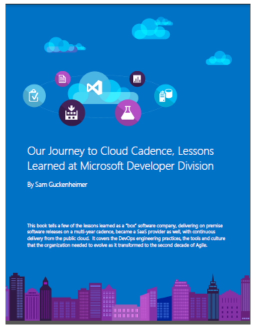
Foreword by Brian Harry, Microsoft Technical Fellow

Visual Studio Team Foundation Server 2012 Adopting Agile Software Practices

From Backlog to Continuous Feedback

Visual Studio

Sam Guckenheimer & Neno Loje



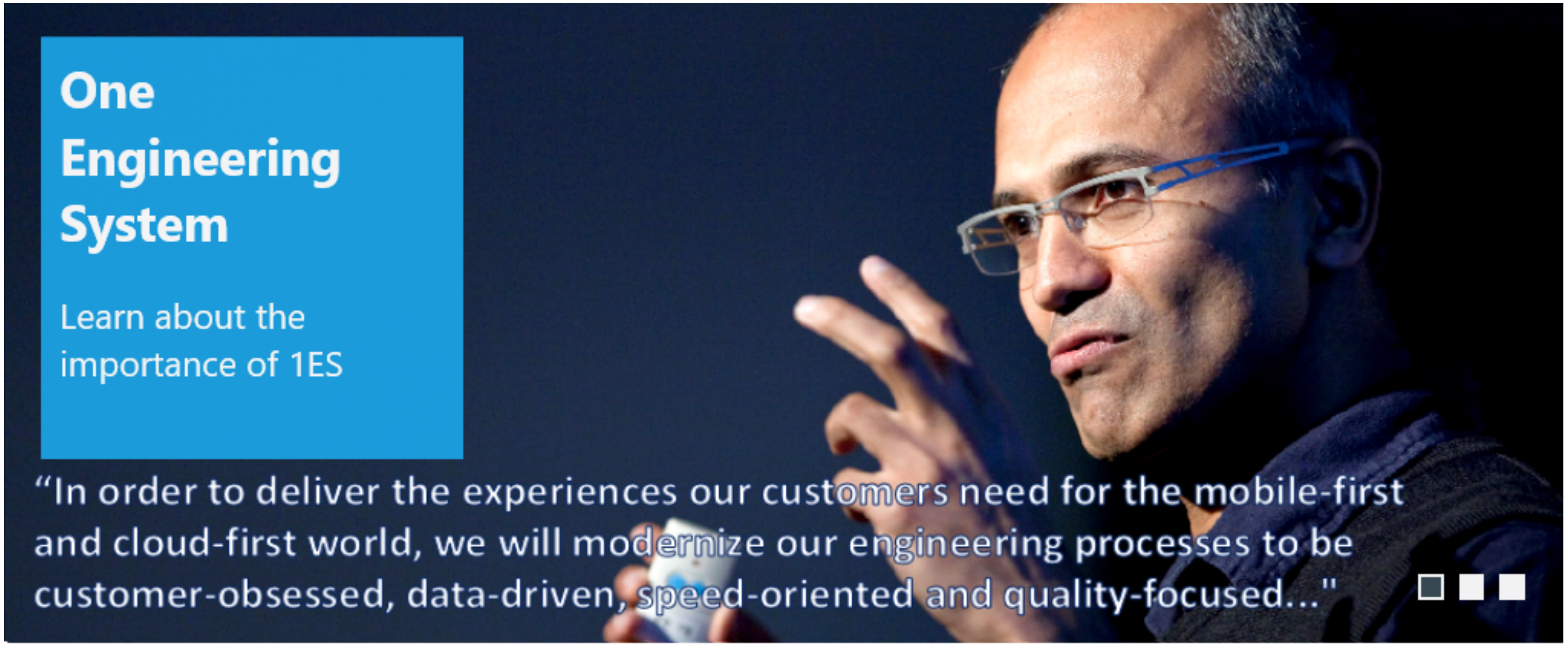
Our Journey to Cloud Cadence, Lessons Learned at Microsoft Developer Division

By Sam Guckenheimer

This book tells a few of the lessons learned as a "box" software company, delivering on premise software releases on a multi-year cadence, became a SaaS provider as well, with continuous delivery from the public cloud. It covers the DevOps engineering practices, the tools and culture that the organization needed to evolve as it transformed to the second decade of Agile.

Purpose of One Engineering System

If you want to go fast, go alone. If you want to go far, go together.



One Engineering System

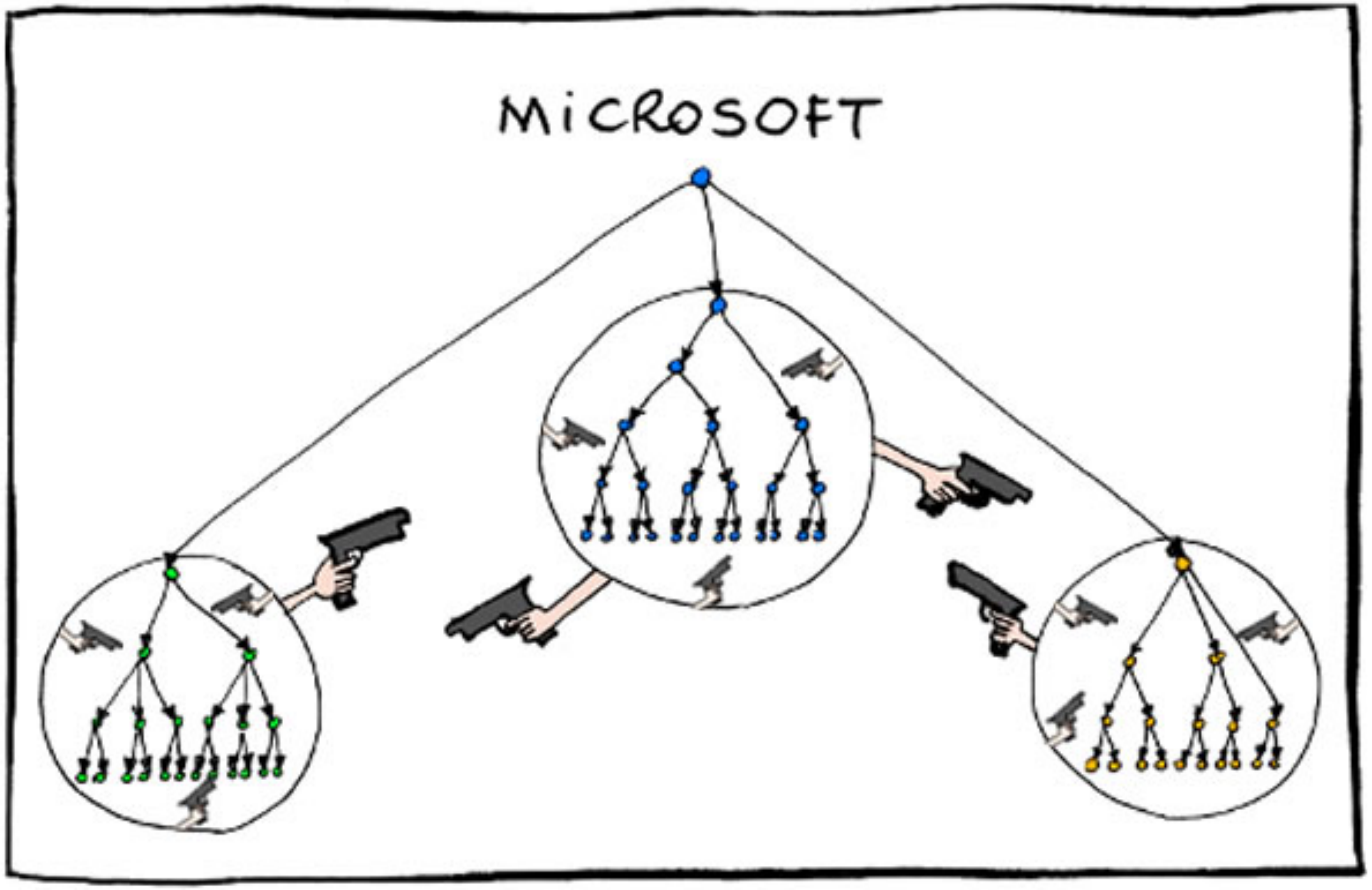
Learn about the importance of 1ES

"In order to deliver the experiences our customers need for the mobile-first and cloud-first world, we will modernize our engineering processes to be customer-obsessed, data-driven, speed-oriented and quality-focused..."

Before 1ES

Unintended consequence:

No reuse would go unpunished.



http://www.bonkersworld.net/images/2011.06.27_organizational_charts.png

An engineering north star…

…the source across the company is available to anyone

…any dev can offer improvements to anything in the company

…the IP the company has built up over the years is made of re-usable components

…anybody can find and potentially re-use components from anywhere else

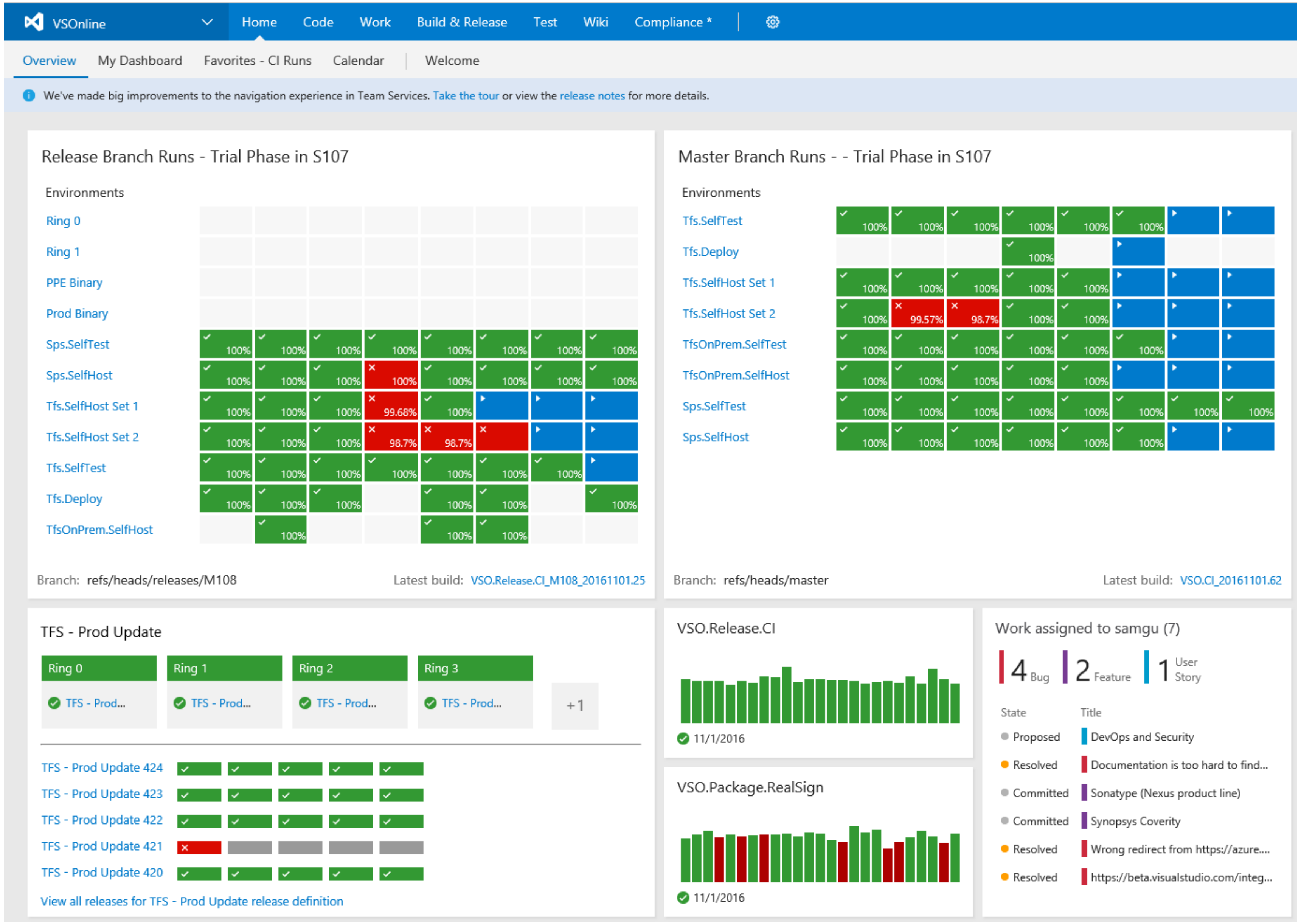
…devs are rewarded for creating popular components

…there is zero lag from when a dev makes a change & when the rest of the company sees it

… build and test time is directly proportional to the change made

…devs can move to another team and already know how to work

What it looks like in practice



VSOnline | Home | Code | Work | Build & Release | Test | Wiki | Compliance *

Overview | My Dashboard | Favorites - CI Runs | Calendar | Welcome

We've made big improvements to the navigation experience in Team Services. Take the tour or view the release notes for more details.

Release Branch Runs - Trial Phase in S107

Environments								
Ring 0								
Ring 1								
PPE Binary								
Prod Binary								
Sps.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.SelfHost	✓ 100%	✓ 100%	✓ 100%	✗ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Tfs.SelfHost Set 1	✓ 100%	✓ 100%	✓ 100%	✗ 99.68%	✓ 100%	▶	▶	▶
Tfs.SelfHost Set 2	✓ 100%	✓ 100%	✓ 100%	✗ 98.7%	✗ 98.7%	✗	▶	▶
Tfs.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶
Tfs.Deploy	✓ 100%	✓ 100%	✓ 100%		✓ 100%	✓ 100%		✓ 100%
TfsOnPrem.SelfHost		✓ 100%			✓ 100%	✓ 100%		

Branch: refs/heads/releases/M108 Latest build: VSO.Release.CI_M108_20161101.25

Master Branch Runs - - Trial Phase in S107

Environments								
Tfs.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶	▶
Tfs.Deploy				✓ 100%		▶		
Tfs.SelfHost Set 1	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶	▶	▶
Tfs.SelfHost Set 2	✓ 100%	✗ 99.57%	✗ 98.7%	✓ 100%	✓ 100%	▶	▶	▶
TfsOnPrem.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶	▶
TfsOnPrem.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶	▶	▶
Sps.SelfTest	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%
Sps.SelfHost	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	✓ 100%	▶	▶

Branch: refs/heads/master Latest build: VSO.CI_20161101.62

TFS - Prod Update

Ring 0	Ring 1	Ring 2	Ring 3	
✓ TFS - Prod...	✓ TFS - Prod...	✓ TFS - Prod...	✓ TFS - Prod...	+1

TFS - Prod Update 424	✓	✓	✓	✓	✓
TFS - Prod Update 423	✓	✓	✓	✓	✓
TFS - Prod Update 422	✓	✓	✓	✓	✓
TFS - Prod Update 421	✗	▨	▨	▨	▨
TFS - Prod Update 420	✓	✓	✓	✓	✓

View all releases for TFS - Prod Update release definition

VSO.Release.CI

✓ 11/1/2016

VSO.Package.RealSign

✓ 11/1/2016

Work assigned to samgu (7)

4 Bug | 2 Feature | 1 User Story

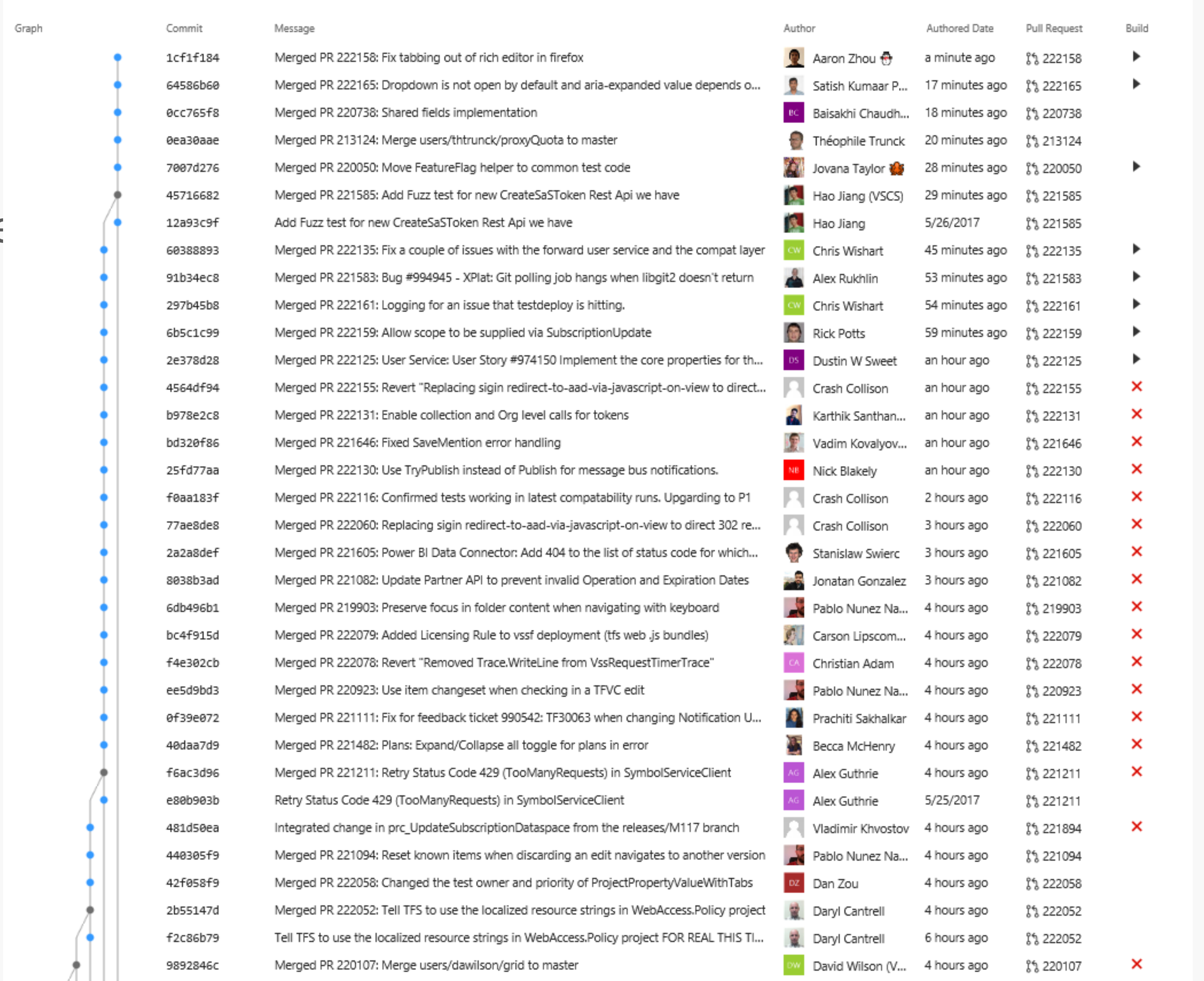
State	Title
● Proposed	DevOps and Security
● Resolved	Documentation is too hard to find...
● Committed	Sonatype (Nexus product line)
● Committed	Synopsys Coverity
● Resolved	Wrong redirect from https://azure....
● Resolved	https://beta.visualstudio.com/integ...

Scaling Git for enterprise rigor

Everyone is on ONE maste branch

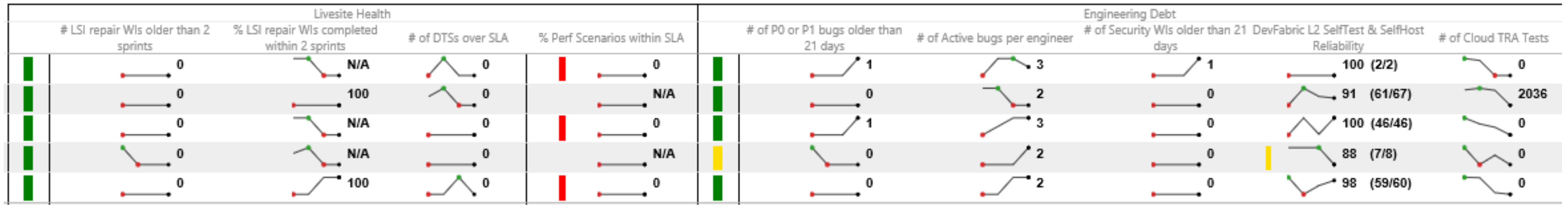
Topic branching allows Pull Request flow with tiny review and merging

Code fresh in your mind



Graph	Commit	Message	Author	Authored Date	Pull Request	Build
	1cf1f184	Merged PR 222158: Fix tabbing out of rich editor in firefox	Aaron Zhou	a minute ago	222158	▶
	64586b60	Merged PR 222165: Dropdown is not open by default and aria-expanded value depends o...	Satish Kumaar P...	17 minutes ago	222165	▶
	0cc765f8	Merged PR 220738: Shared fields implementation	Baisakhi Chaudh...	18 minutes ago	220738	
	0ea30aae	Merged PR 213124: Merge users/thtrunck/proxyQuota to master	Théophile Trunck	20 minutes ago	213124	
	7007d276	Merged PR 220050: Move FeatureFlag helper to common test code	Jovana Taylor	28 minutes ago	220050	▶
	45716682	Merged PR 221585: Add Fuzz test for new CreateSaSToken Rest Api we have	Hao Jiang (VSCS)	29 minutes ago	221585	
	12a93c9f	Add Fuzz test for new CreateSaSToken Rest Api we have	Hao Jiang	5/26/2017	221585	
	60388893	Merged PR 222135: Fix a couple of issues with the forward user service and the compat layer	Chris Wishart	45 minutes ago	222135	▶
	91b34ec8	Merged PR 221583: Bug #994945 - XPlat: Git polling job hangs when libgit2 doesn't return	Alex Rukhlin	53 minutes ago	221583	▶
	297b45b8	Merged PR 222161: Logging for an issue that testdeploy is hitting.	Chris Wishart	54 minutes ago	222161	▶
	6b5c1c99	Merged PR 222159: Allow scope to be supplied via SubscriptionUpdate	Rick Potts	59 minutes ago	222159	▶
	2e378d28	Merged PR 222125: User Service: User Story #974150 Implement the core properties for th...	Dustin W Sweet	an hour ago	222125	▶
	4564df94	Merged PR 222155: Revert "Replacing sigin redirect-to-aad-via-javascript-on-view to direct...	Crash Collison	an hour ago	222155	✗
	b978e2c8	Merged PR 222131: Enable collection and Org level calls for tokens	Karthik Santhan...	an hour ago	222131	✗
	bd320f86	Merged PR 221646: Fixed SaveMention error handling	Vadim Kovalyov...	an hour ago	221646	✗
	25fd77aa	Merged PR 222130: Use TryPublish instead of Publish for message bus notifications.	Nick Blakely	an hour ago	222130	✗
	f0aa183f	Merged PR 222116: Confirmed tests working in latest compatability runs. Upgarding to P1	Crash Collison	2 hours ago	222116	✗
	77ae8de8	Merged PR 222060: Replacing sigin redirect-to-aad-via-javascript-on-view to direct 302 re...	Crash Collison	3 hours ago	222060	✗
	2a2a8def	Merged PR 221605: Power BI Data Connector: Add 404 to the list of status code for which...	Stanislaw Swierc	3 hours ago	221605	✗
	8038b3ad	Merged PR 221082: Update Partner API to prevent invalid Operation and Expiration Dates	Jonatan Gonzalez	3 hours ago	221082	✗
	6db496b1	Merged PR 219903: Preserve focus in folder content when navigating with keyboard	Pablo Nunez Na...	4 hours ago	219903	✗
	bc4f915d	Merged PR 222079: Added Licensing Rule to vssf deployment (tfs web .js bundles)	Carson Lipscom...	4 hours ago	222079	✗
	f4e302cb	Merged PR 222078: Revert "Removed Trace.WriteLine from VssRequestTimerTrace"	Christian Adam	4 hours ago	222078	✗
	ee5d9bd3	Merged PR 220923: Use item changeset when checking in a TFVC edit	Pablo Nunez Na...	4 hours ago	220923	✗
	0f39e072	Merged PR 221111: Fix for feedback ticket 990542: TF30063 when changing Notification U...	Prachiti Sakhalkar	4 hours ago	221111	✗
	40daa7d9	Merged PR 221482: Plans: Expand/Collapse all toggle for plans in error	Becca McHenry	4 hours ago	221482	✗
	f6ac3d96	Merged PR 221211: Retry Status Code 429 (TooManyRequests) in SymbolServiceClient	Alex Guthrie	4 hours ago	221211	✗
	e80b903b	Retry Status Code 429 (TooManyRequests) in SymbolServiceClient	Alex Guthrie	5/25/2017	221211	
	481d50ea	Integrated change in prc_UpdateSubscriptionDataspace from the releases/M117 branch	Vladimir Khvostov	4 hours ago	221894	✗
	440305f9	Merged PR 221094: Reset known items when discarding an edit navigates to another version	Pablo Nunez Na...	4 hours ago	221094	
	42f058f9	Merged PR 222058: Changed the test owner and priority of ProjectPropertyValueWithTabs	Dan Zou	4 hours ago	222058	
	2b55147d	Merged PR 222052: Tell TFS to use the localized resource strings in WebAccess.Policy project	Daryl Cantrell	4 hours ago	222052	
	f2c86b79	Tell TFS to use the localized resource strings in WebAccess.Policy project FOR REAL THIS TI...	Daryl Cantrell	6 hours ago	222052	
	9892846c	Merged PR 220107: Merge users/dawilson/grid to master	David Wilson (V...	4 hours ago	220107	✗

Live Site Culture and Engineering



	Livesite Health					Engineering Debt						
	# LSI repair WIs older than 2 sprints	% LSI repair WIs completed within 2 sprints	# of DTSs over SLA		% Perf Scenarios within SLA		# of P0 or P1 bugs older than 21 days	# of Active bugs per engineer	# of Security WIs older than 21 days		DevFabric L2 SelfTest & SelfHost Reliability	# of Cloud TRA Tests
🟩	0	N/A	0	🟥	0	🟩	1	3	1		100 (2/2)	0
🟩	0	100	0		N/A	🟩	0	2	0		91 (61/67)	2036
🟩	0	N/A	0	🟥	0	🟩	1	3	0		100 (46/46)	0
🟩	0	N/A	0		N/A	🟨	0	2	0	🟨	88 (7/8)	0
🟩	0	100	0	🟥	0	🟩	0	2	0		98 (59/60)	0

Live Site Health

Time to Detect

Time to Communicate

Time To Mitigate

Customer Impact

Incident prevention items

Aging live site problems

Customer support metrics

SLA per customer account

(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 by test level

Usage

Acquisition

Engagement

Dedication

Churn

Feature usage

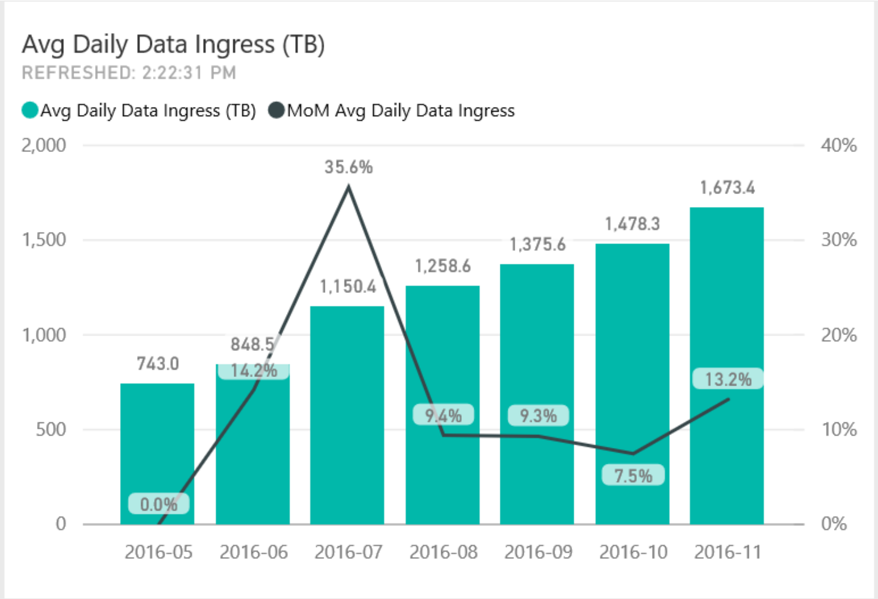
Common Telemetry Pipeline

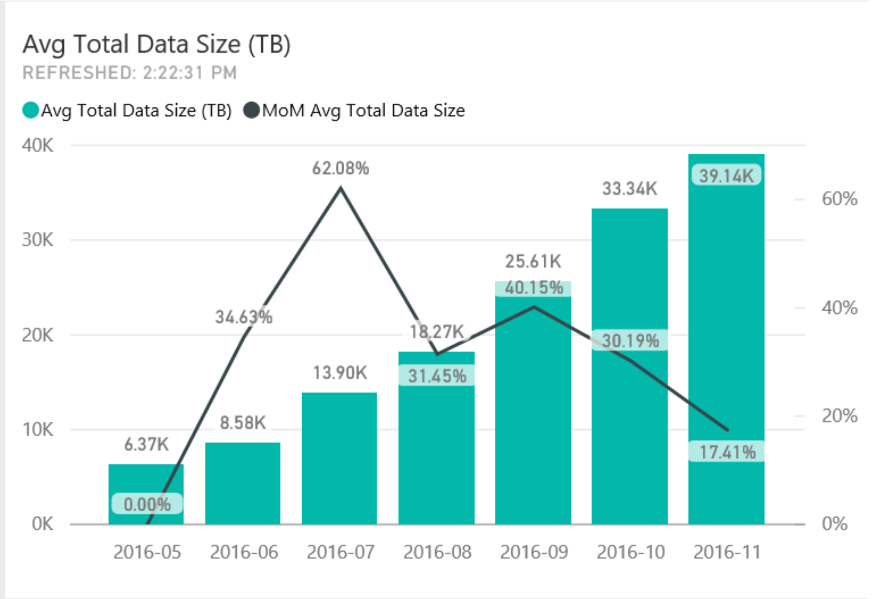
Combines

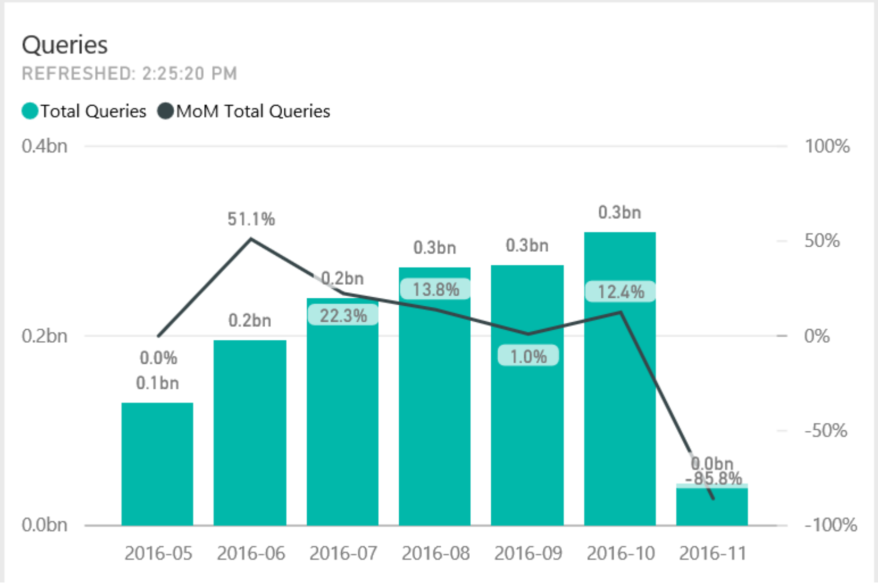
high volume ingestion

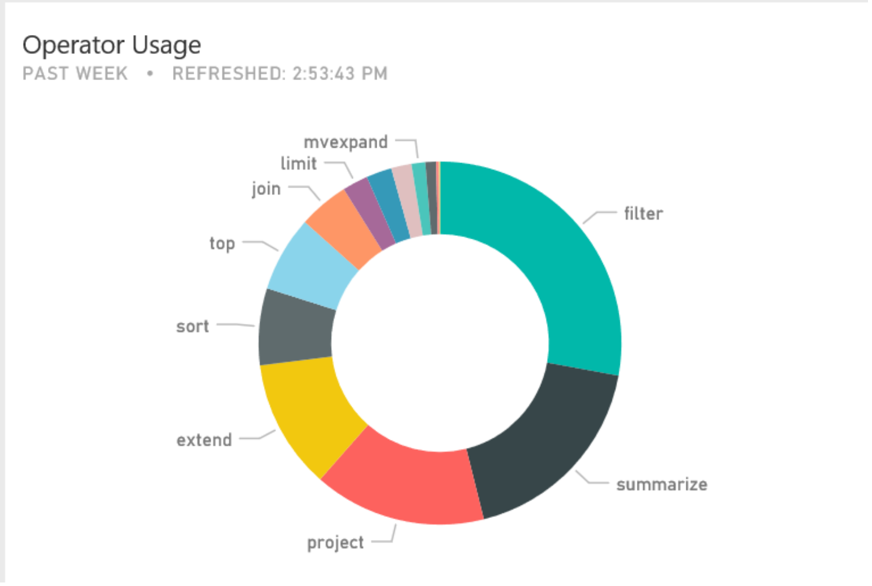
fast queries over very large data sets

text search structured and unstructured data









We Create 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.

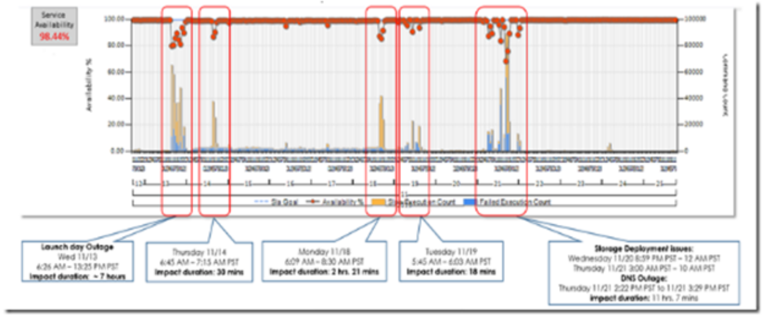
We've had some issues with the service over the past week and a half. I feel terrible about it and I can't apologize enough. It's the biggest incident we've had since the instability created by our service refactoring in the March/April timeframe. I know it's not much consolation but I can assure you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

RATE THIS

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

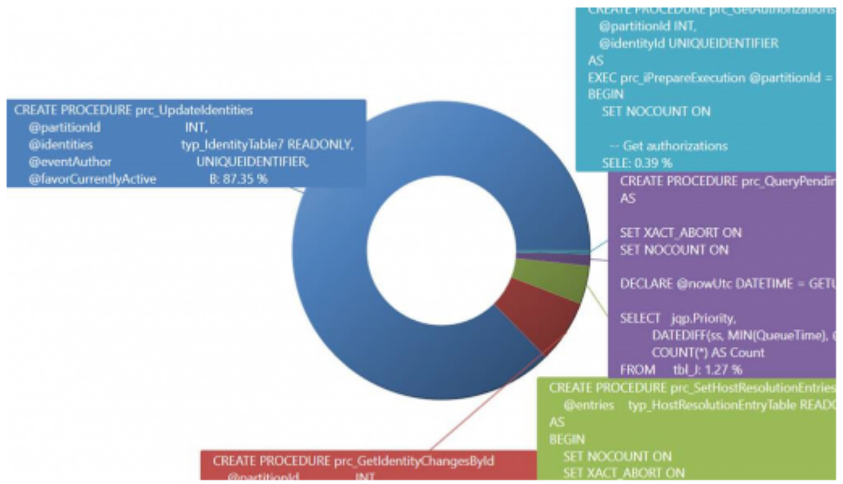
I'd say the first and foremost lesson is "Don't let a 'nice to have' feature take down your mission critical ones." There's a notion in services that all services should be loosely coupled and failure tolerant. One service going down should not cause a cascading failure, causing other services to fail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

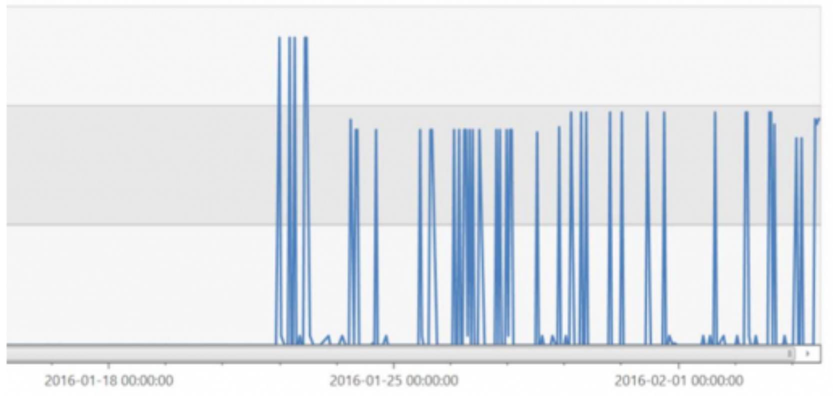
02/06/2016 by Brian Harry MS // 15 Comments

f 0 0 in 0

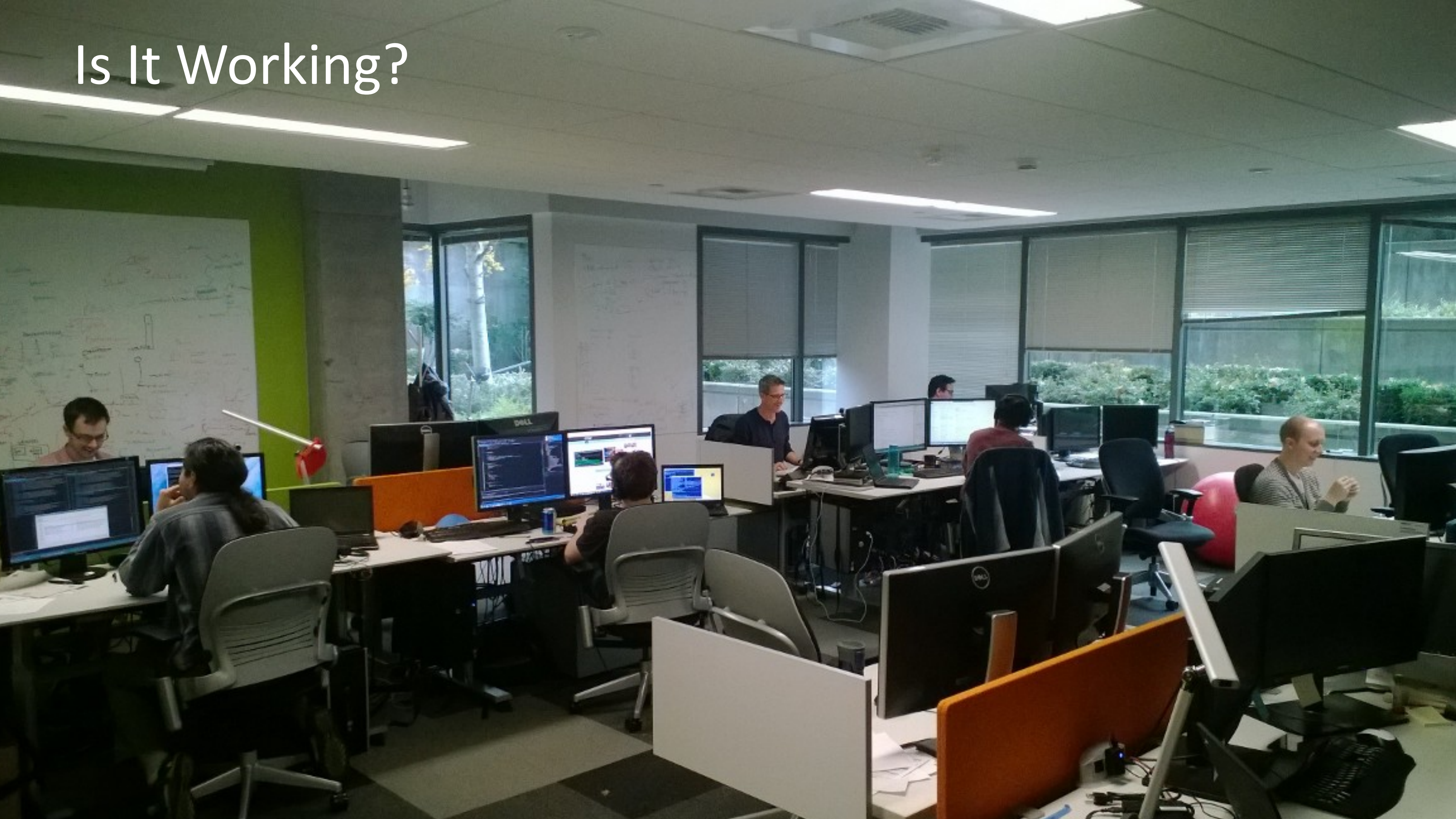
Drilling further by looking at what sprocs are waiting on RESOURCE_SEMAPHORE, we see that prc_UpdateIdentities dominates. Guess what… That's the sproc that caused this incident.



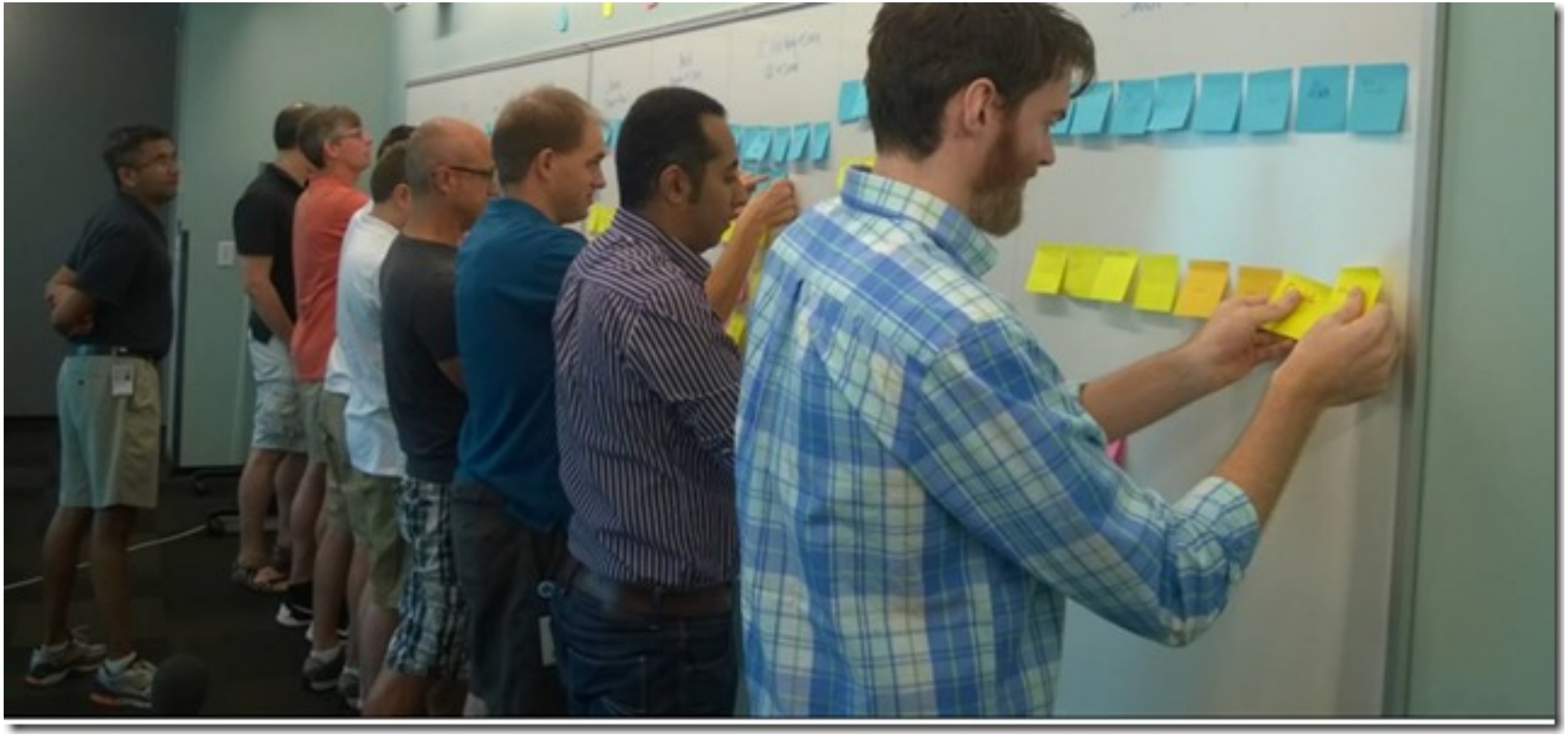
And now, let's look at a time chart of memory grant requests for this sproc. The huge spikes begin the moment we introduced the change to SQL compat level. This is a fantastic opportunity for automated anomaly detection. There's no reason we can't find this kind of thing long before it creates any actual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



Is It Working?



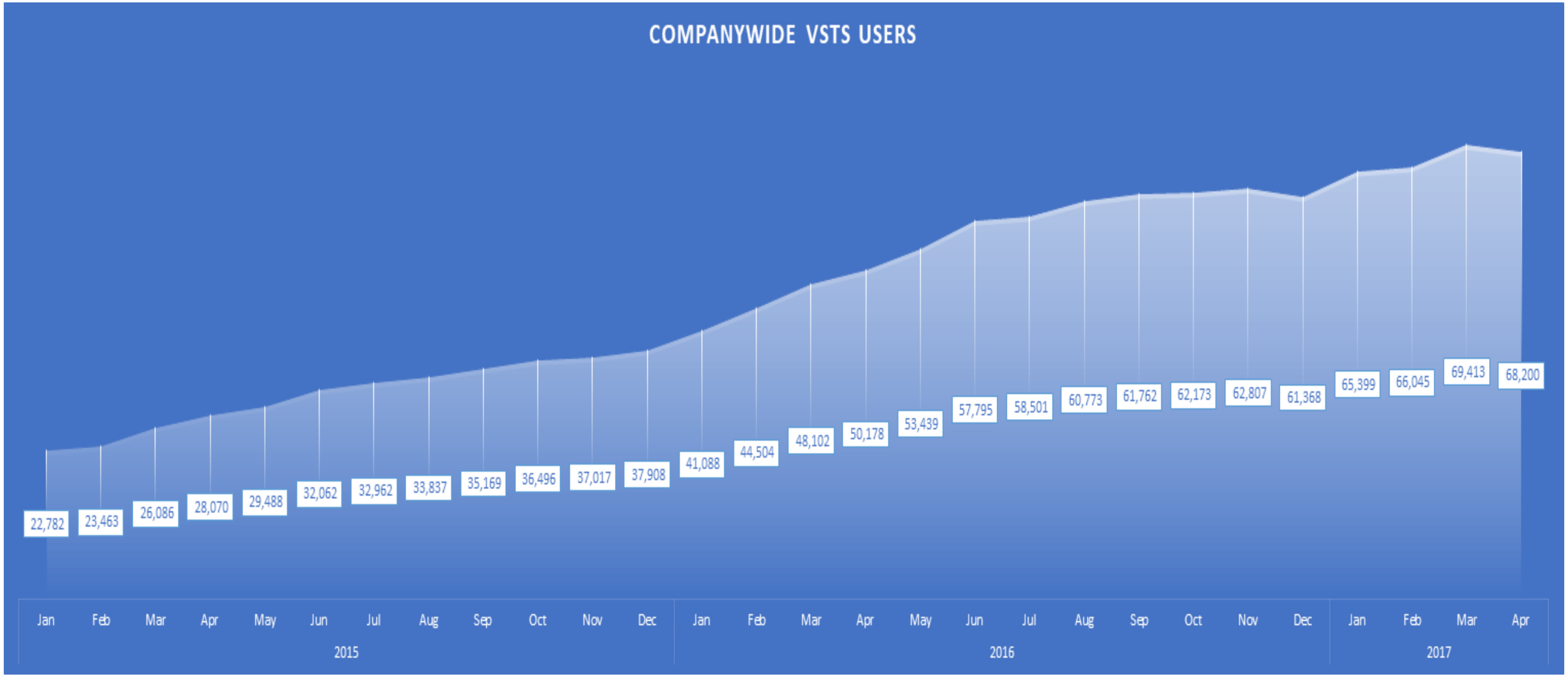
Self-forming teams



https://aka.ms/SelfFormingTeams

Is 1ES Working?

4x active user growth in 2 years to 68,000


COMPANYWIDE VSTS USERS
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec