



Strategic Architect Forum 2015

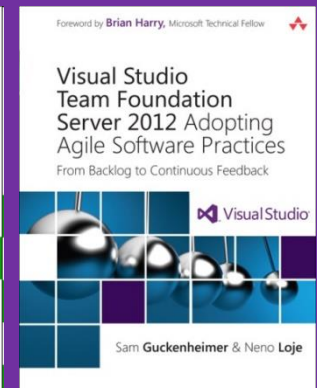
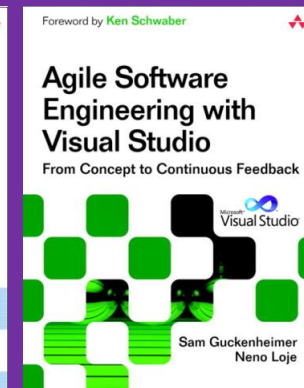
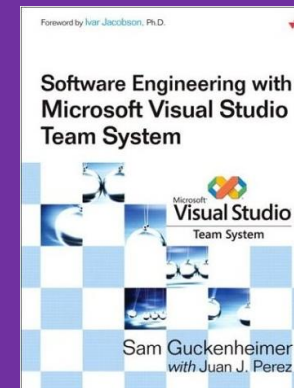
January 27th–28th, 2015 • Bellevue, WA – Westin

Session code:

DevOps: the Second Decade of Agile

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

Sam Guckenheimer
Product Owner, Visual Studio Product Line
Microsoft Corporation
[@samguckenheimer](https://twitter.com/samguckenheimer)
[@shitsamgusays](https://twitter.com/shitsamgusays)



In the beginning...



2005: a culture of silos

CONWAY'S LAW

The best and brightest

Autonomy, job rotation,
promotion

The currency of love

Headcount

DYSFUNCTIONAL TRIBALISM

Don't ask, don't tell

Schedule chicken

Metrics are for others

Our tribe is better

Our customers are different

WASTE

Easy credit

No interest charge for debt

PRIMARY COMMUNICATION

Blame game

Unintended consequence: waste

MUDA	In-process inventory	Bug debt and incomplete work carried forward. Requires multiple handling, creates overhead, creates stress
	Over-production	Peanut butter. Teams create low-pri features and make them self-justifying. This work squeezes capacity from the high-pri work
	Extra processing	Bug debt, reactivations, triage, redundant testing, relearning of others' code, handling broken dependencies
	Transportation	Handoffs across roles, teams, divisions, etc.
	Motion	Managing enlistments, lab setup, parallel release work
	Waiting	Delays, blocking bugs, incomplete incoming
	Unevenness	Lumpy work breakdown clogging the flow
MURA	Inconsistency	Different definitions of "done"
	Absurdity	Stress due to excessive scope
MURI	Unreasonableness	Expectations of heroics
	Overburden	Stress due to excessive overhead

Unintended consequence: delay

3–4 YEARS TO BUILD IT

~~M0~~: plan and cost the release

M1...M3: develop the code

err...M3.1...M3.3 recode

Beta1: integrate and pray

Beta2: test like hell

RC (release candidate) 0..n: final builds

RTM: ship it!

SERVICE IT (10 YEARS)

QFEs and hot fixes

Service packs

BUSINESS MODEL: ONE-TIME LICENSE

Unintended consequence: tech debt

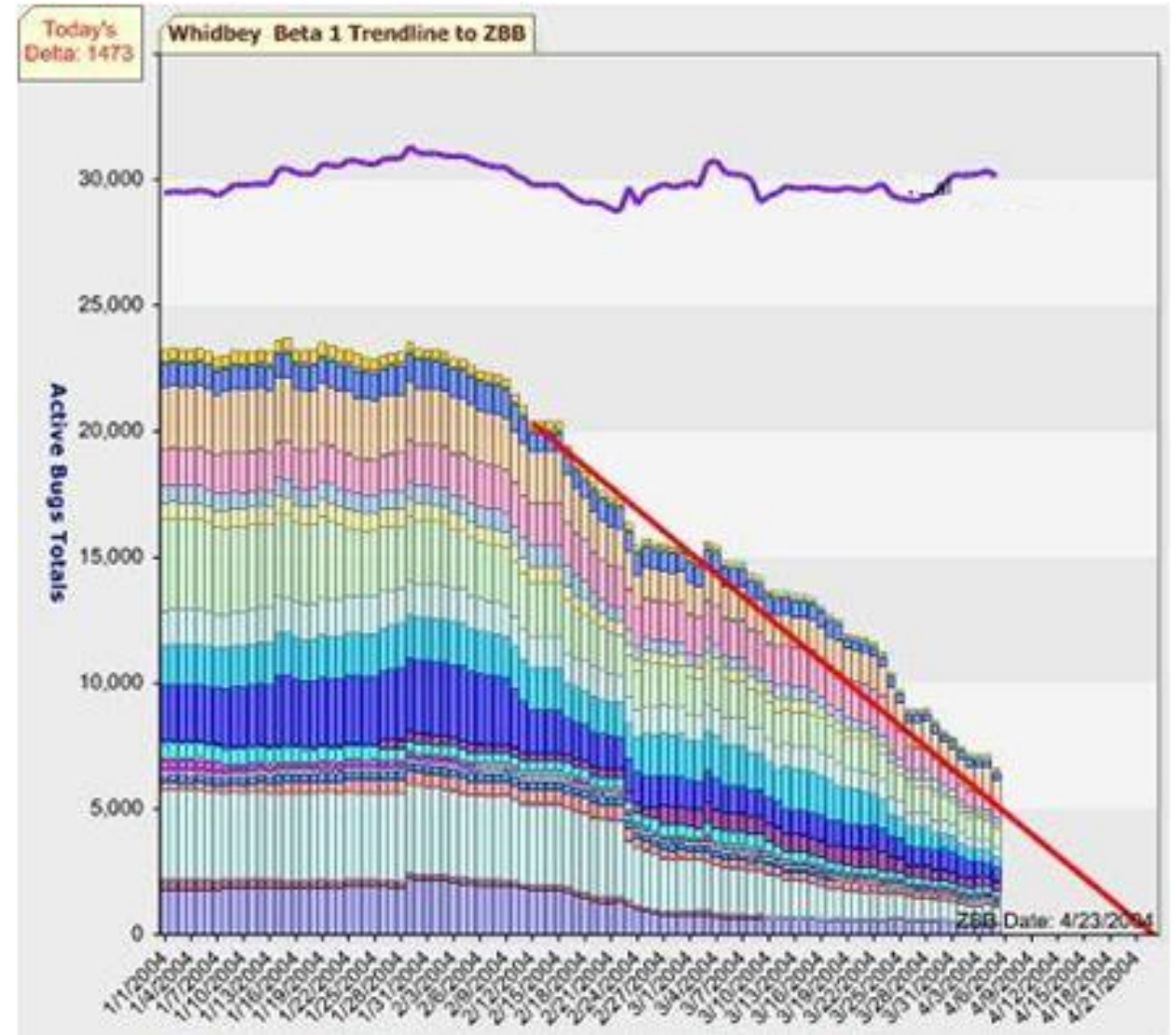
A.k.a. In-process inventory + extra processing

Visual Studio 2005 Bug Step @Beta 1

Deferral a common technique
(see top line)

Teams carry undone work
(both functional and 'ility)

Endgame hard to predict



Adopting Agile

Adopting Agile practices

Get clean, stay clean	Product backlog	Defining done	Scheduling
Iteration backlog	Engineering principles	Measurement and hardening	Results

Changes to pre- and post game

BUILD IT (2 YEARS)

Planning: start and groom product backlog

MQ: improve our practices, get ready for the next version

M0: What are our goals for this release? What customer value do we deliver?

M1...M3 5 week sprints: develop and test the code

CTP (customer technical preview): targeted customer release to collect feedback on mainline scenarios

Beta1: first broad customer visibility; collect feedback

Beta2: validate recent changes with customers; collect feedback

RC (release candidate) 0..n: final builds

RTM: ship it!

SERVICE IT (10 YEARS)

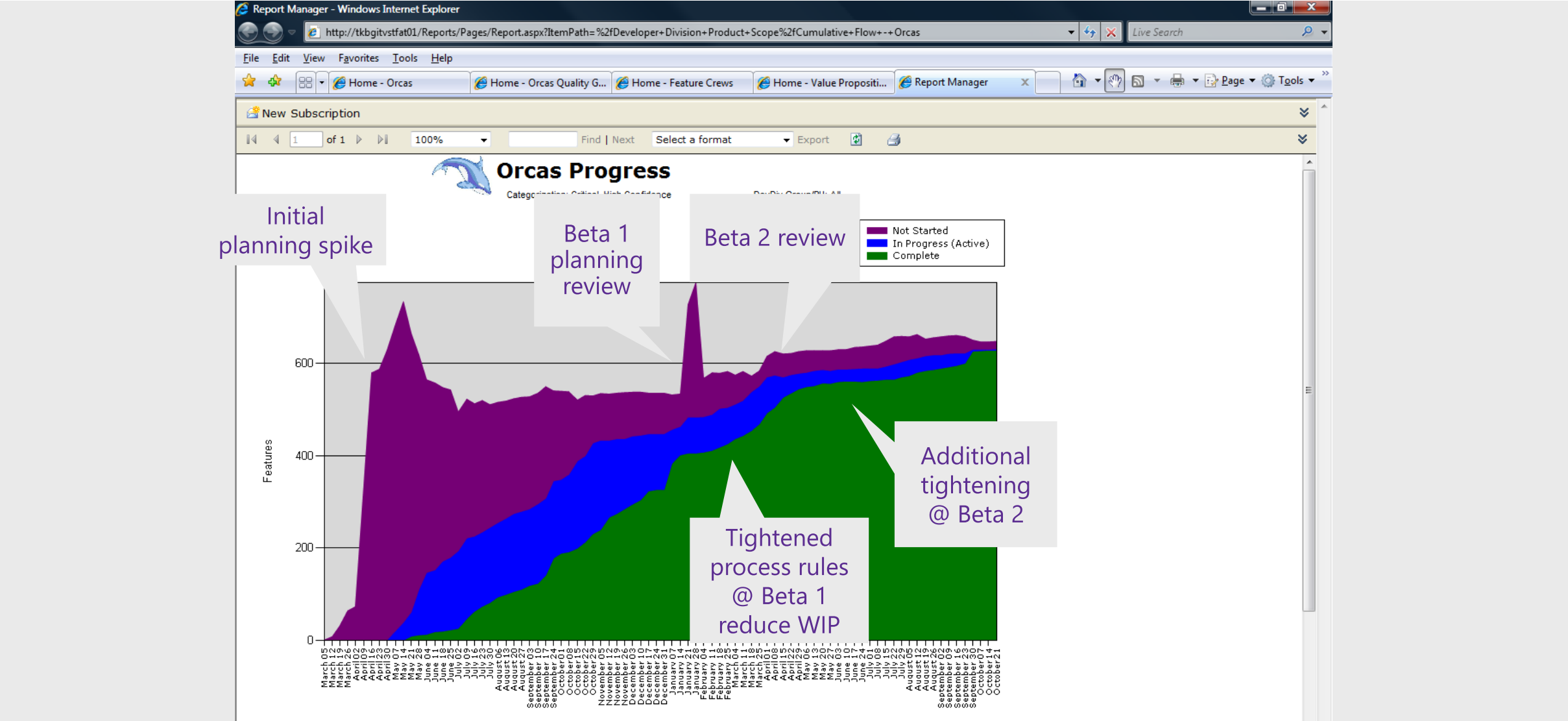
Quarterly: power tools and feature packs for current release

Business transformation from packaged product to subscription

Common practices

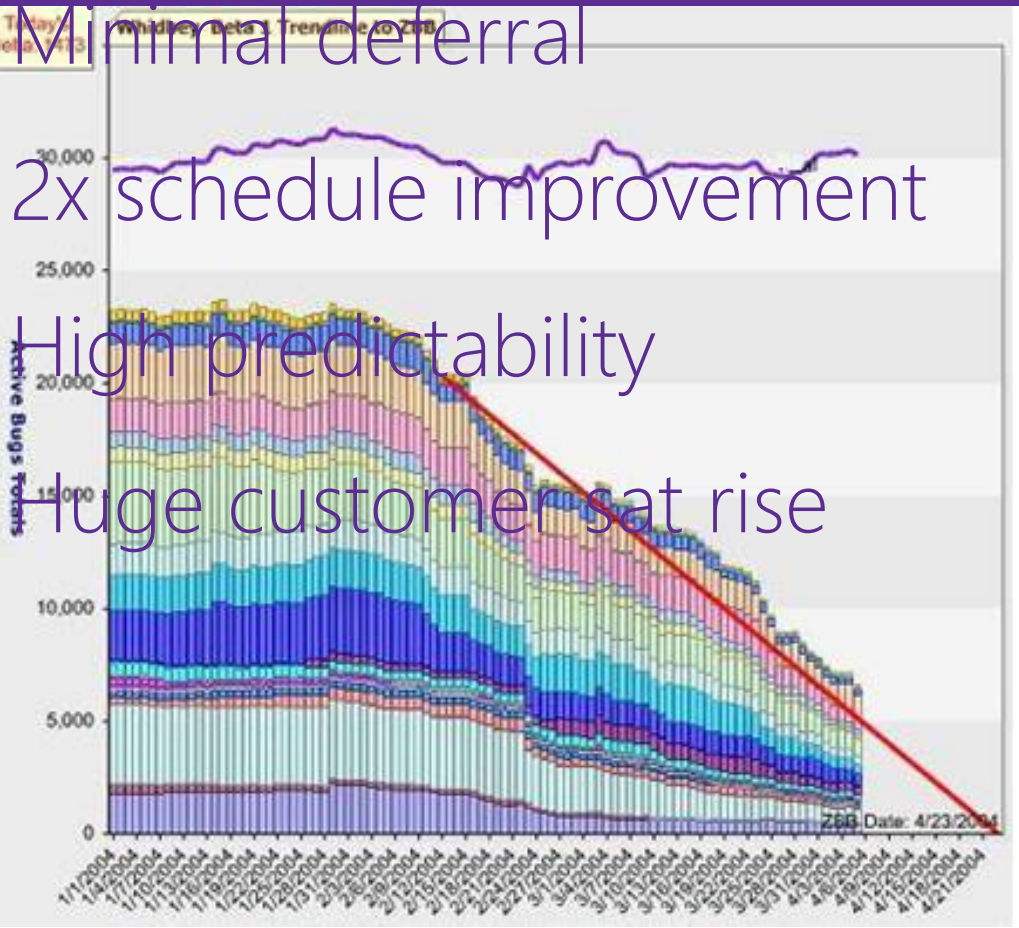
Get clean, stay clean	Product backlog	Defining done	Scheduling
Iteration backlog	Engineering principles	Measurement and hardening	Results

Actual feature flow in Orcas (Visual Studio 2008)

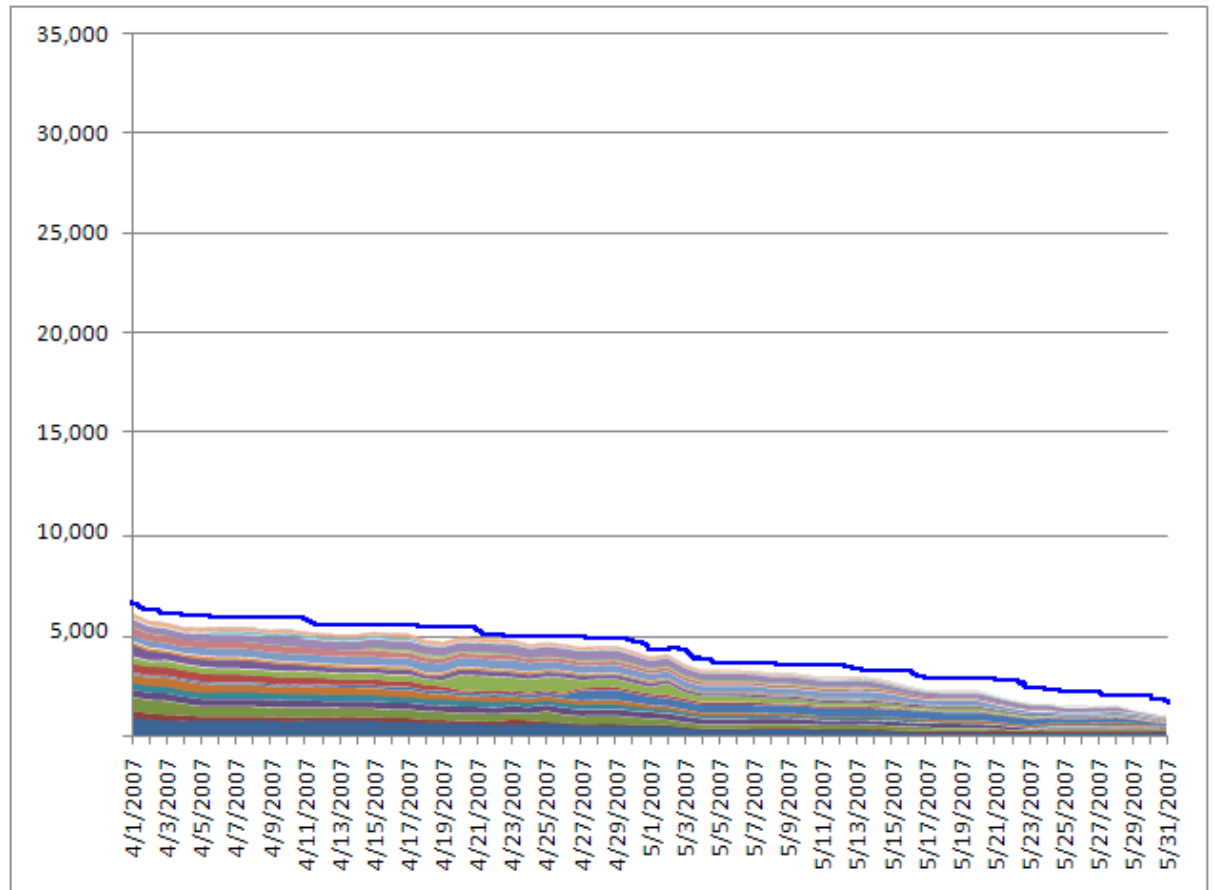


2005 debt versus 2008 debt

VISUAL STUDIO 2005 BETA 1: PRODUCT BUGS ONLY



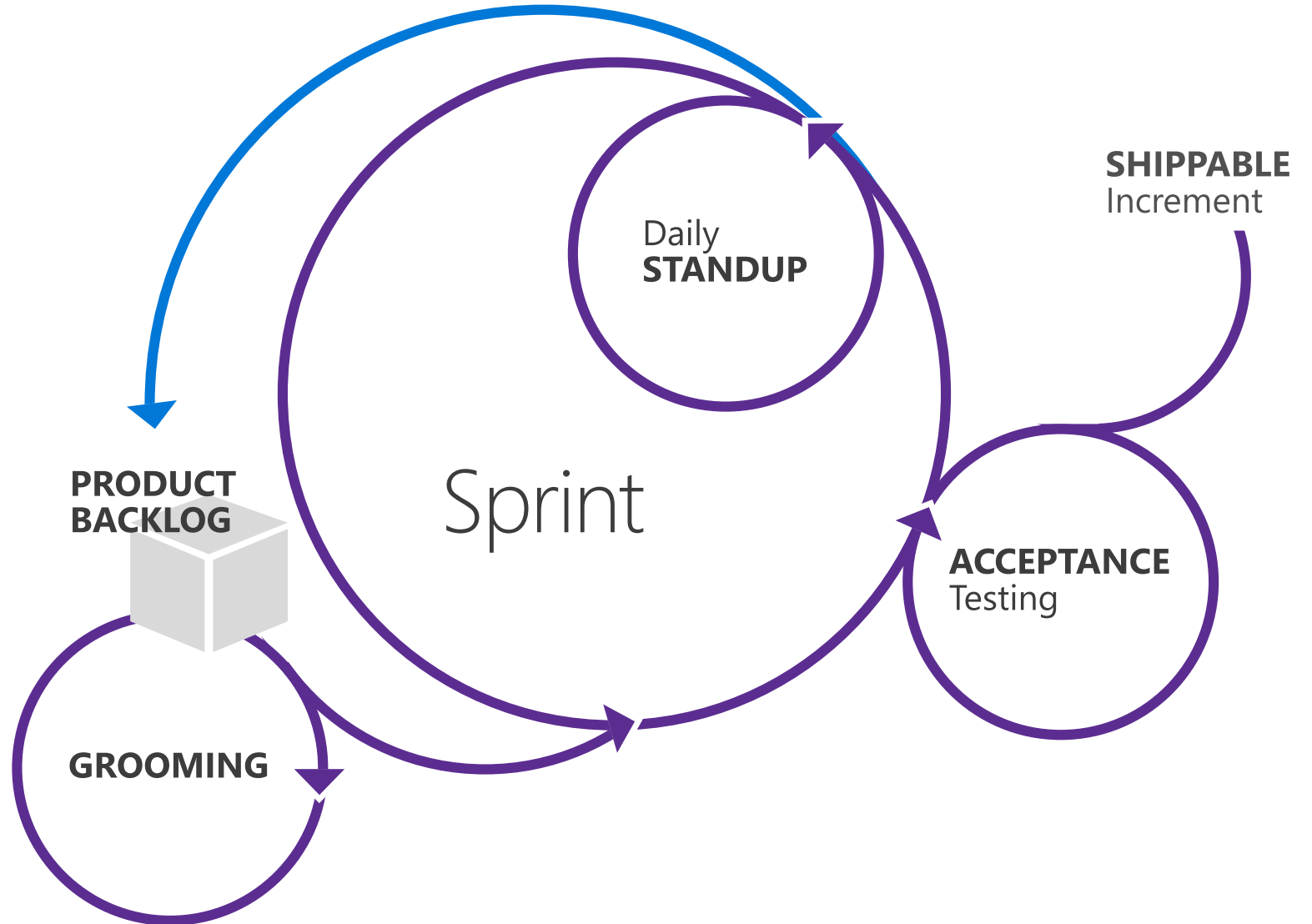
VISUAL STUDIO 2008 BETA 1: ALL BUG DEBT



Unintended consequences

Internally focused

Is the backlog
correctly ranked?





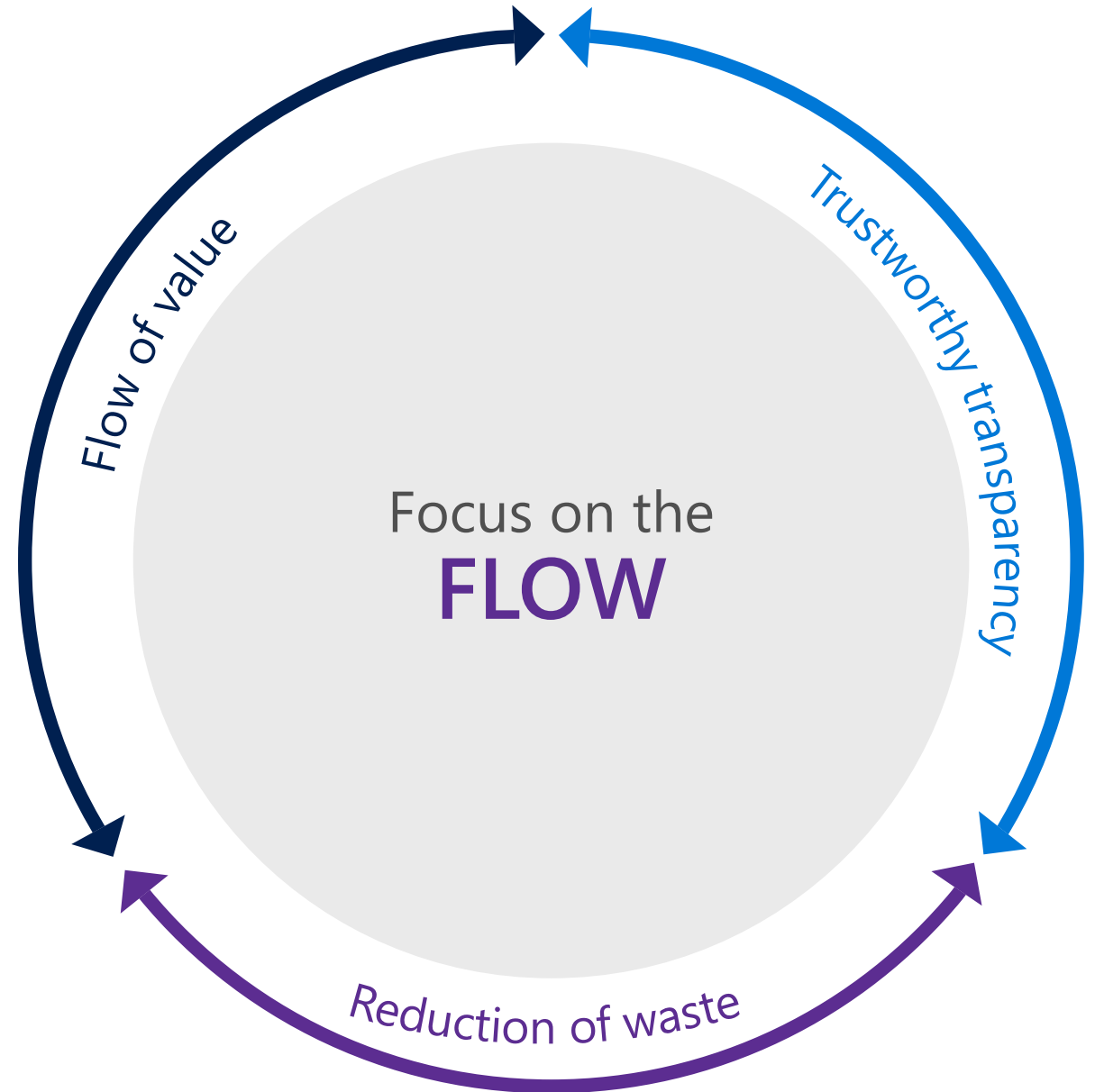
- ### Agenda
- Votel-ALM Scenarios
 - Discussion
 - Make Developers Raving Fans
 - Cloud Hosting
 - Agile Teams
 - Developer -> Testing -> Test-Labs Lifecycle
 - Requirements
 - Automatable Data
- Visual Studio

HOW DO WE GATHER CUSTOMER FEEDBACK?



THE VOTES ARE VISIBLE, BUT THE PROCESS IS HARD TO SCALE AND REPEAT

Having removed debt,
we could focus on
flow of value



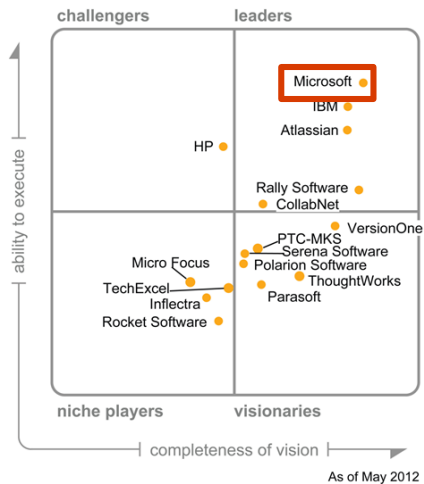
We achieved a great position

COMMUNITY VOTES FOR BEST PRODUCT

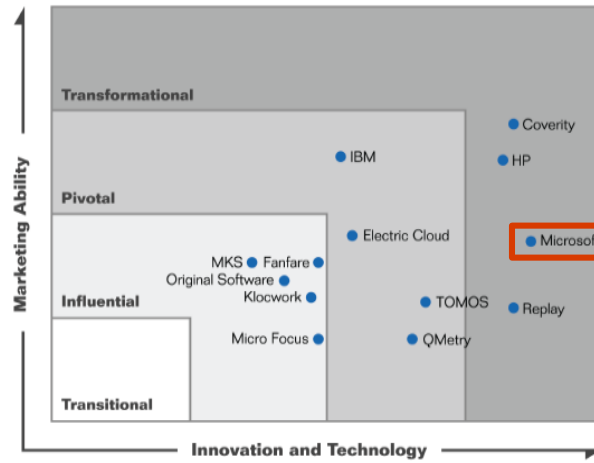


INDUSTRY ANALYSTS RECOGNIZE MICROSOFT LEADERSHIP

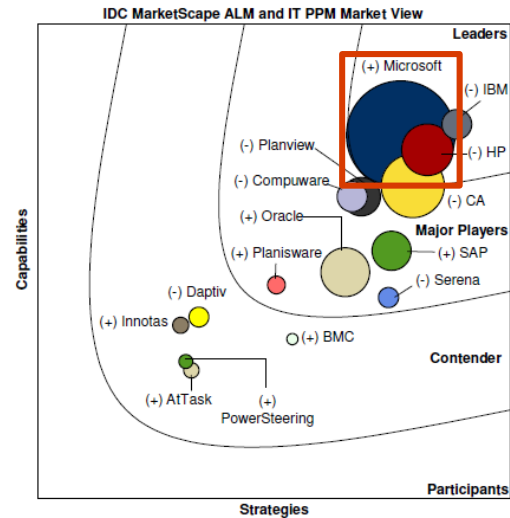
Figure 1. Magic Quadrant for Application Life Cycle Management



GARTNER:
**Magic Quadrant
 for Application
 Life Cycle
 Management 5
 June 2012**



VOKE:
**Test Market
 Mover Array
 July 2010
 Evaluating Visual
 Studio2010**



IDC:
**IT PPM Market
 Landscape
 December 2010
 Evaluating Visual
 Studio2010**



USAGE CULTURE

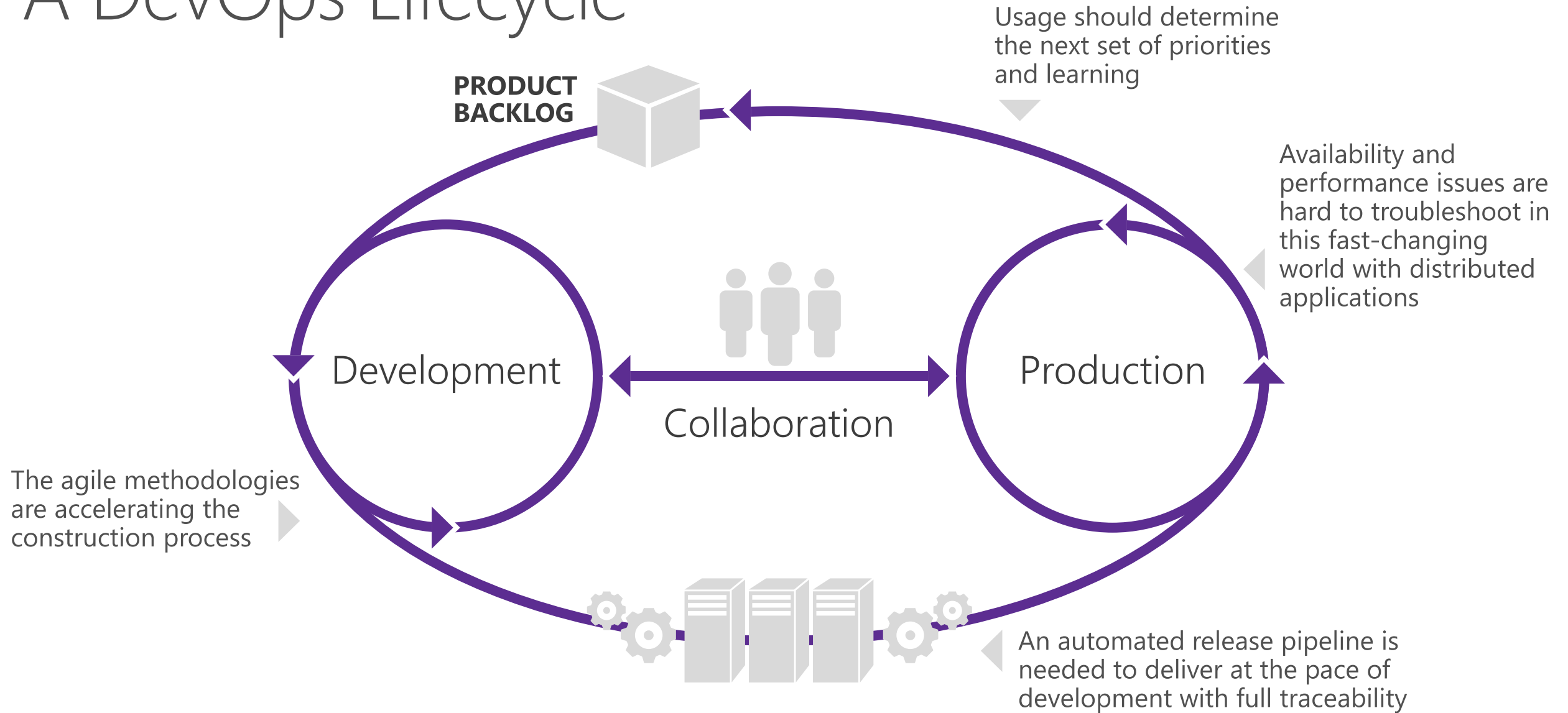
Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

A DevOps Lifecycle



Example: Visual Studio Online evolves features that are later shared with on premise

The screenshot shows the Visual Studio Online interface for an Agile Release. The top navigation bar includes 'Visual Studio Online / VSONline / Agile Release' and the user 'Aaron Bjork'. Below the navigation, there are tabs for 'HOME', 'CODE', 'WORK', 'BUILD', and 'TEST', with 'Overview' selected. A search bar for work items is also present.

The main content area is titled 'Sprint 69' for the period 'July 14 - August 1'. It displays a progress bar showing '163 of 175 h' and 'Stories: 4 not started, 6 in progress'. A line chart shows the sprint's progress over time.

On the right, there are links for 'Work' (Backlog, Task board, Queries), 'Visual Studio' (Open in Visual Studio, Get Visual Studio), and 'Team rooms' (Agile Release Room with 2 users in room). A 'Members' section shows a grid of team members and a total count of 16.

The 'Pinned Items' section contains several charts:

- Active Bugs - Distribution:** A donut chart showing the distribution of active bugs among team members. The total number of bugs is 33. The distribution is: (blank) 18, Andrew Olson 1, Bob Bao (Pa...) 1, Pranav Kark... 2, Peter Kruger 3, Liang Zhu 3, Nianwen So... 3, Sam Nuziale 3, and (other) 3.
- Active Bugs - Trend:** A stacked area chart showing the trend of active bugs from 6/28/2014 to 7/22/2014. The legend indicates counts of 0, 3, 1, and 2 for different categories.
- DTS over time:** A line chart showing DTS (Defects To Solve) over time for two categories: '5 - PG Eng...' and '4 - PG Tri...'. The current value is 2.
- Triage Query:** A large blue card showing 0 Work items.
- Active DTS:** A large blue card showing 1 Work item.

Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Scrum teams (a.k.a. “feature crews”)

Cross-discipline

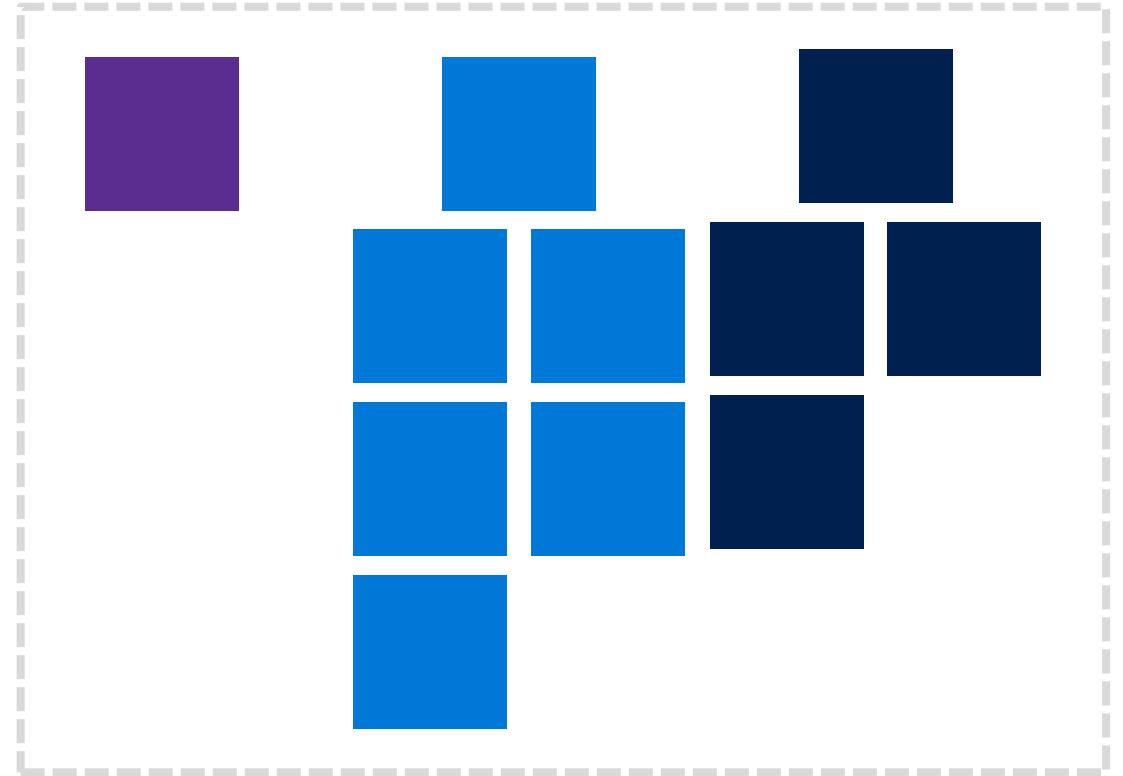
8–12 people

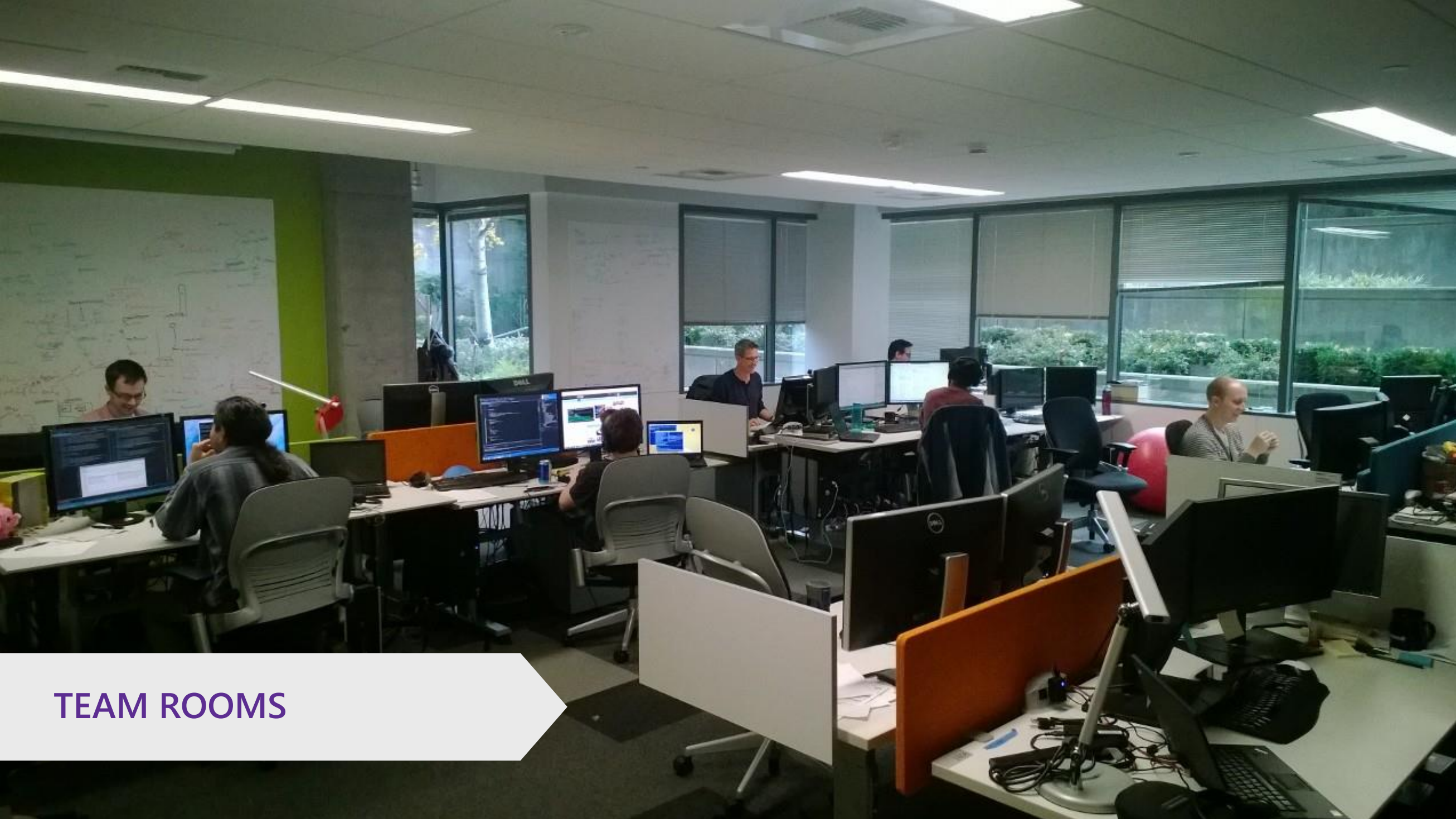
Led by their triad

Intact for 12–18 months

Autonomous backlog

Sample area: version control





TEAM ROOMS

Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Planning horizons

SPRINT

3-week sprints

3

PLANNING CHAT

3 sprints

3

SEASON

6-month season

6

VISION

18-month vision

18

Three-week sprints

SPRINT 56

WEEK 1

WEEK 2

WEEK 3

Start planning

Done!

Deployment

SPRINT 57

WEEK 1

WEEK 2

WEEK 3

SPRINT 58

WEEK 1

WEEK 2

WEEK 3

Low ceremony: sprint mails and video

SPRINT 56

WEEK 1

WEEK 2

WEEK 3



The sprint plan



What we accomplished

SPRINT 57

WEEK 1

WEEK 2

WEEK 3

SPRINT 58

WEEK 1

WEEK 2


WEEK 3


Sprint mails: example


Collaboration - Sprint 41 Plan - Message (HTML)

FILE MESSAGE

Mon 12/3/2012 12:42 PM

 David Gorena 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 Gorena 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

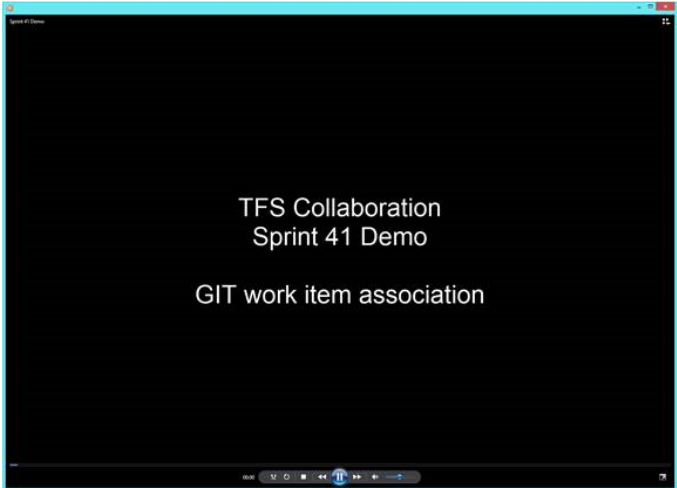
 You forwarded this message on 1/7/2013 8:54 AM.


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



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

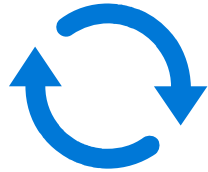
Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Deployment principles



ALWAYS UP



FULLY
AUTOMATED



DECOUPLED
SERVICES WITH
CLEAR CONTRCTS



FEATURE FLAGS



INCREMETAL
(CANARY)
DEPLOYMENT

Feature flag pattern

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 and refinement

Frequent updates

The image shows two browser windows from the Team Foundation Service website. The left window displays a news article titled "Team Foundation Service updates" for September 2013, featuring "Work Item Charts". An inset image shows a screenshot of the TFS interface with a donut chart for a "Product Backlog" showing 16 items (12 Product Backlog Items and 4 Bugs). The right window shows a "Release Archive" table listing updates from 2012 to 2013.

Team Foundation Service updates

2013

- October 21
- October 17
- September 30
- September 9**
- August 19
- July 29
- July 10
- June 26
- June 19
- June 3
- May 28
- May 13
- April 25
- March 22
- March 4
- February 11
- January 30
- January 21
- January 9
- January 7

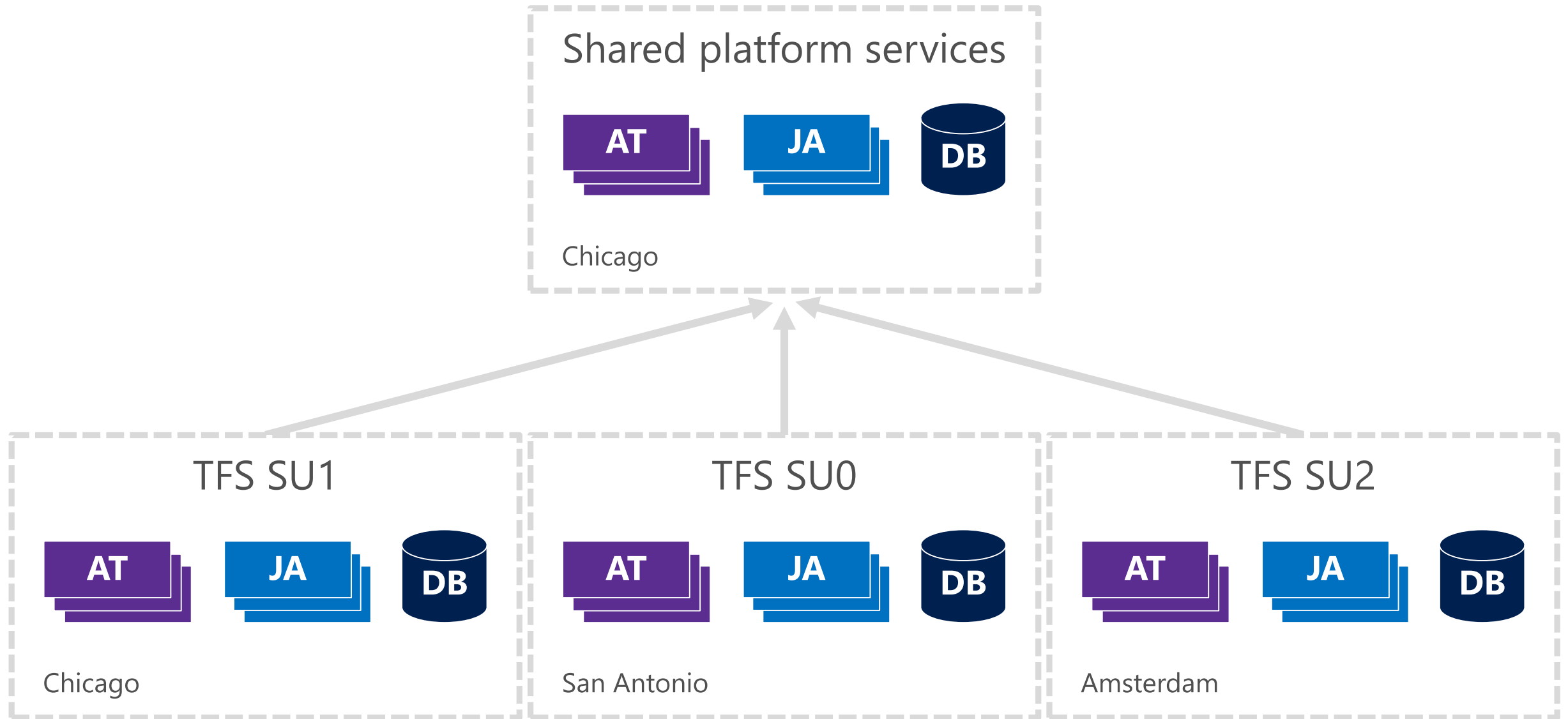
2012

- December 10
- November 19
- October 31
- October 29
- October 8
- September 17
- August 27
- August 13
- August 6

Release Archive

Service	Feature	Server
21 Oct 2013	New account and project pages	2013.1
17 Oct 2013	Build images updated to VS 2013	--
30 Sep 2013	New languages supported for code syntax highlighting	2013.1
	Color picking in charts	2013.1
	Column options for the test case grid view	2013.1
9 Sep 2013	Work item charts	2013
	Bulk edit of test cases	2013
	Delete a team project	2013
19 Aug 2013	Work items from code discussion	2013
	Improved code commenting	2013
	Improved permission management for Git repos	2013
29 Jul 2013	Team room Git push events	2013
	Deleting team rooms	2013
	Backlog mapping	2013
10 Jul 2013	Team permission changes	2013
	Windows 8.1 support in hosted build	--
	Paste images into work items in the web	2013
26 Jun 2013	Open Microsoft Test Runner from web	2013
	Agile Portfolio Management updates - view filter and quick decompose	2013
	Open MTM from web	2013
19 Jun 2013	Admin panel color change	2013
	Agile Portfolio Management	2013
	Lightweight code commenting	2013
3 Jun 2013	Team Room	2013

Multiple scale units/DCs enable canarying



Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Live site culture principles

Site status is
always Pri 1

Global
response team

On-call DRI
(designated
responsible
individual)
by area

Weekly live
site review

Monthly
service review

Live site issues
become
product
backlog items



DASHBOARD ALWAYS VISIBLE

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: SD Bugs - SPS, TFS QA Remaini...

Team favorites: SD Bugs - SPS, TFS QA Remaini...

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		INCIDENT TIMELINE		LIVE SITE REVIEW	
Created Date:	1/10/2014 1:08:25 PM	Incident Start Time:	1/10/2014 1:06:25 PM	Customer Impact:	Yes
Service Category:	Applnsights	Detected:	1/10/2014 1:09:00 PM	Live Site Review?:	Yes
Impacted Service:	GSM	Triaged - VSOLS:	1/10/2014 1:11:00 PM	LSR Owner:	David Goddard
Incident State:	5 - Closed	Escalated - SE:	1/10/2014 1:13:00 PM	LSR Owning Role:	2 - Ops
Severity:	1 - Severe Impact (> 1% to 80% customer impz)	Acknowledged - SE:	1/10/2014 1:13:00 PM	Partner Id:	
Assigned To:	David Goddard	Partner Engaged:		Repeat issue?:	No
Detection Method:	Automated	Engaged - SE:	1/10/2014 1:13:00 PM	Resolution type:	Resolved By SE
Detection Source:	Applnsights	Incident End Time:	1/10/2014 2:54:00 PM	KB ID#:	1396574
Environment:	Production	External Communication:	1/10/2014 1:36:00 PM	Error Category:	Application
Datacenter:		Internal Communication:	1/10/2014 1:34:00 PM	Error Source:	Maintenance
Instance Name:		SE-Effort(hrs):		Error Source SubCategory:	
		SD-Team Count:		Alert count:	

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

MITIGATION

B / U / Bold / Italic / Link / Unlink / Undo / Redo / Print / Refresh / Close

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

Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Driving business success



**"TOP CUSTOMER"
PROGRAM**



CUSTOMER CONTACT



**MONTHLY SERVICE
REVIEW**



**EXPERIMENTATION, OR
BUILD-MEASURE-LEARN**

Every service has multiple funnels

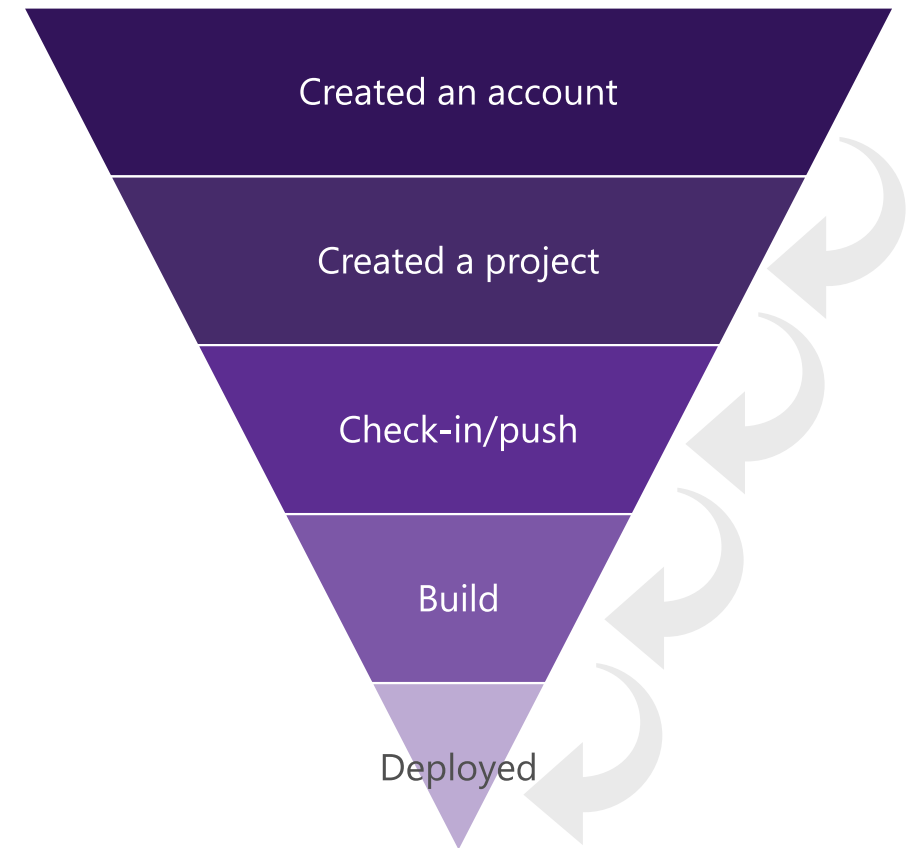
Feeding the mouth of the funnel might be a vanity metric, if conversion rates decrease

Think of each progression and its hassle map to expand funnel efficiency

Experiment!

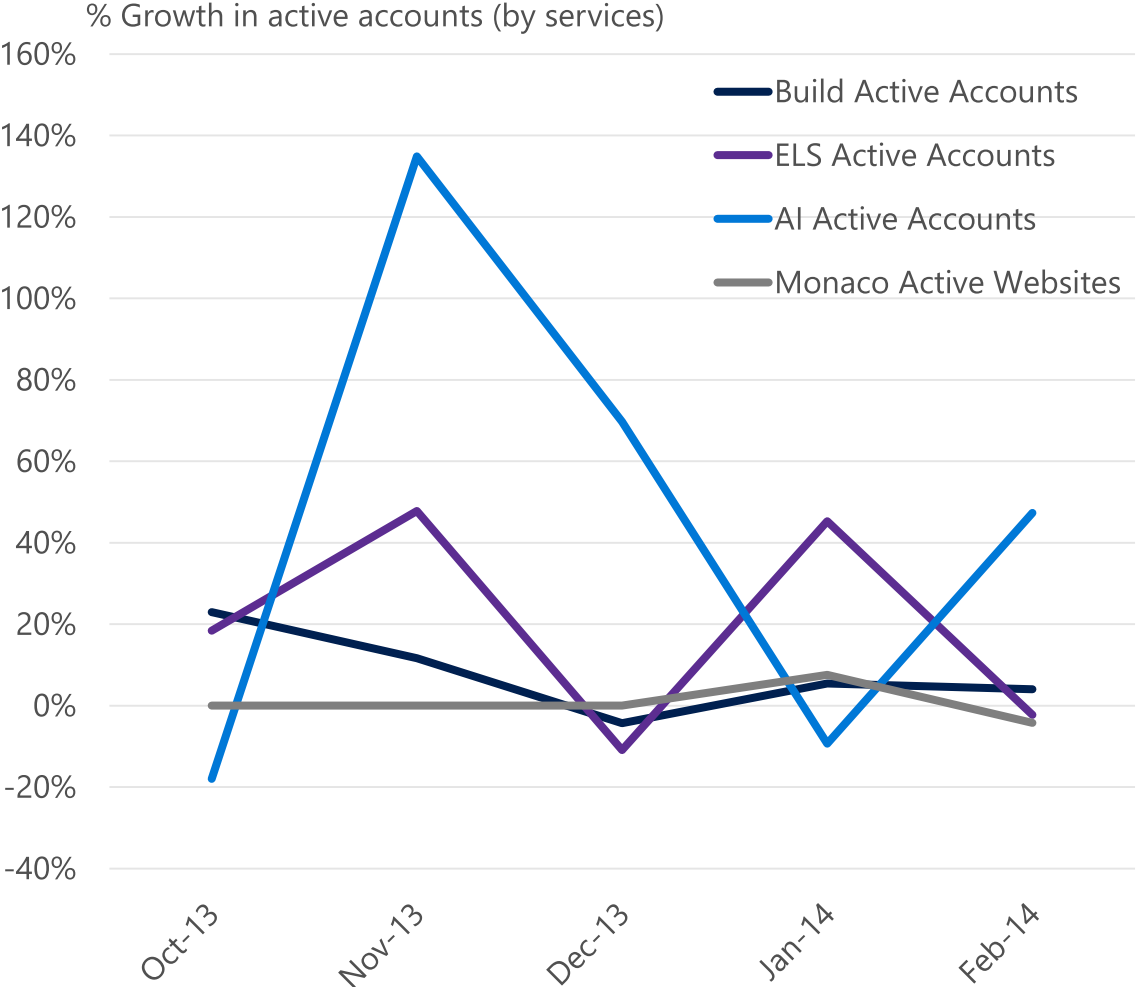
	August	September	October	November	December	January*
Accounts created						
Projects created						
Code checked-in						
Builds executed						
Deployments to Azure						

	August	September	October	November	December	January*
Account → project						
Project → check-in						
Check-in → build						
Build → deploy						
E2E conversion						

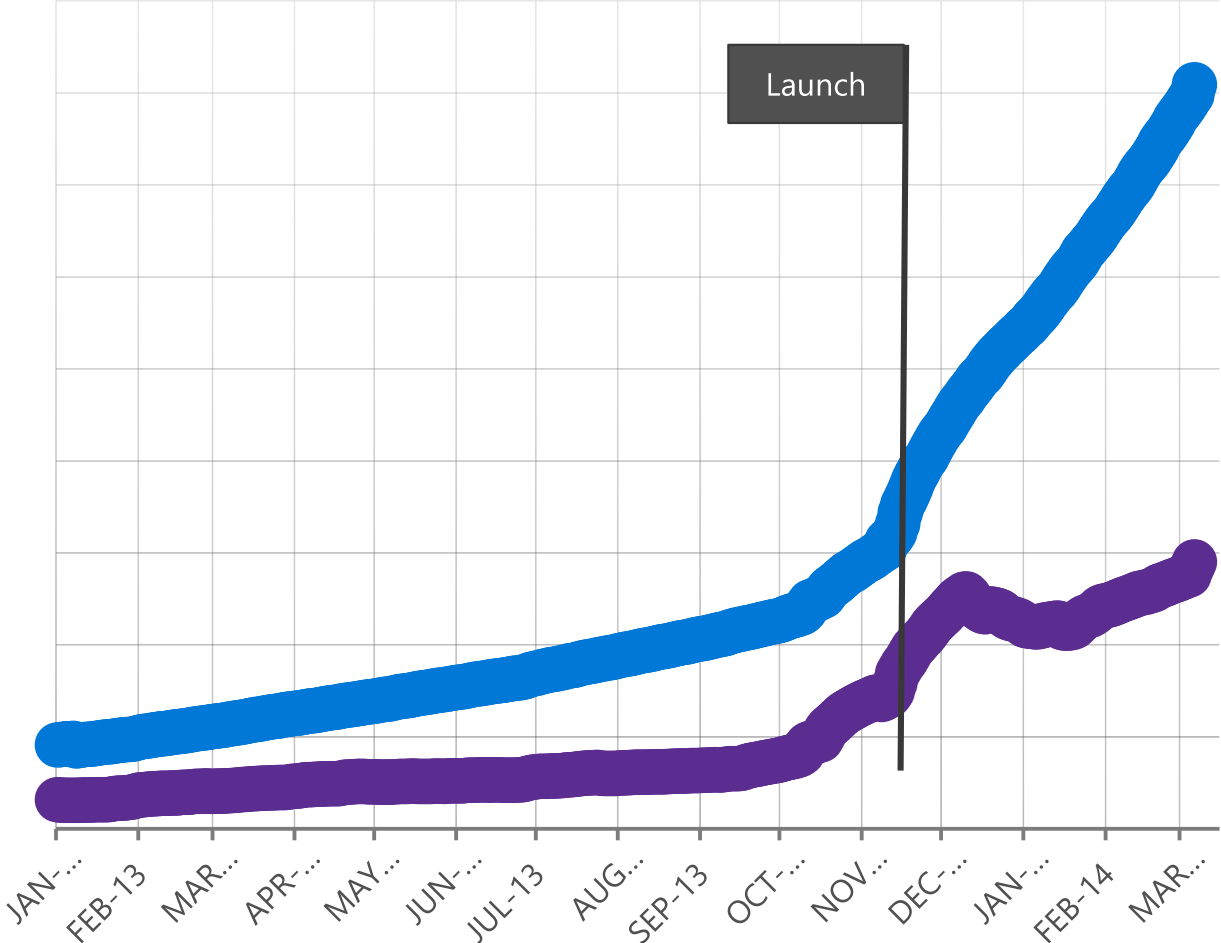


Track MoM growth rates (not just cumulative)

MOM GROWTH IN [SUB]SERVICE ADOPTION



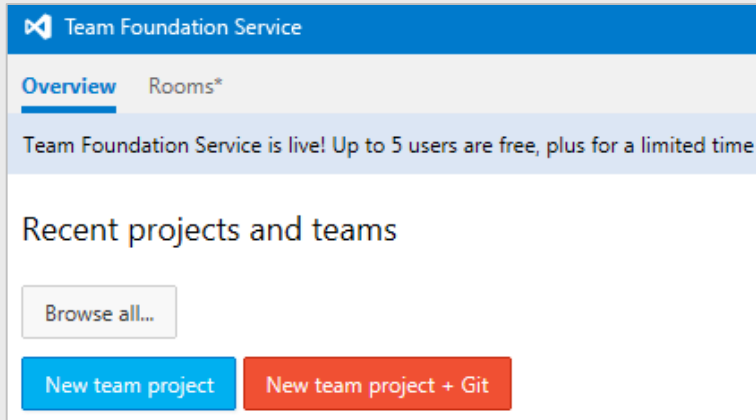
Visual Studio account growth over time



Build, measure, learn

BUILD

We had this...



MEASURE

30%

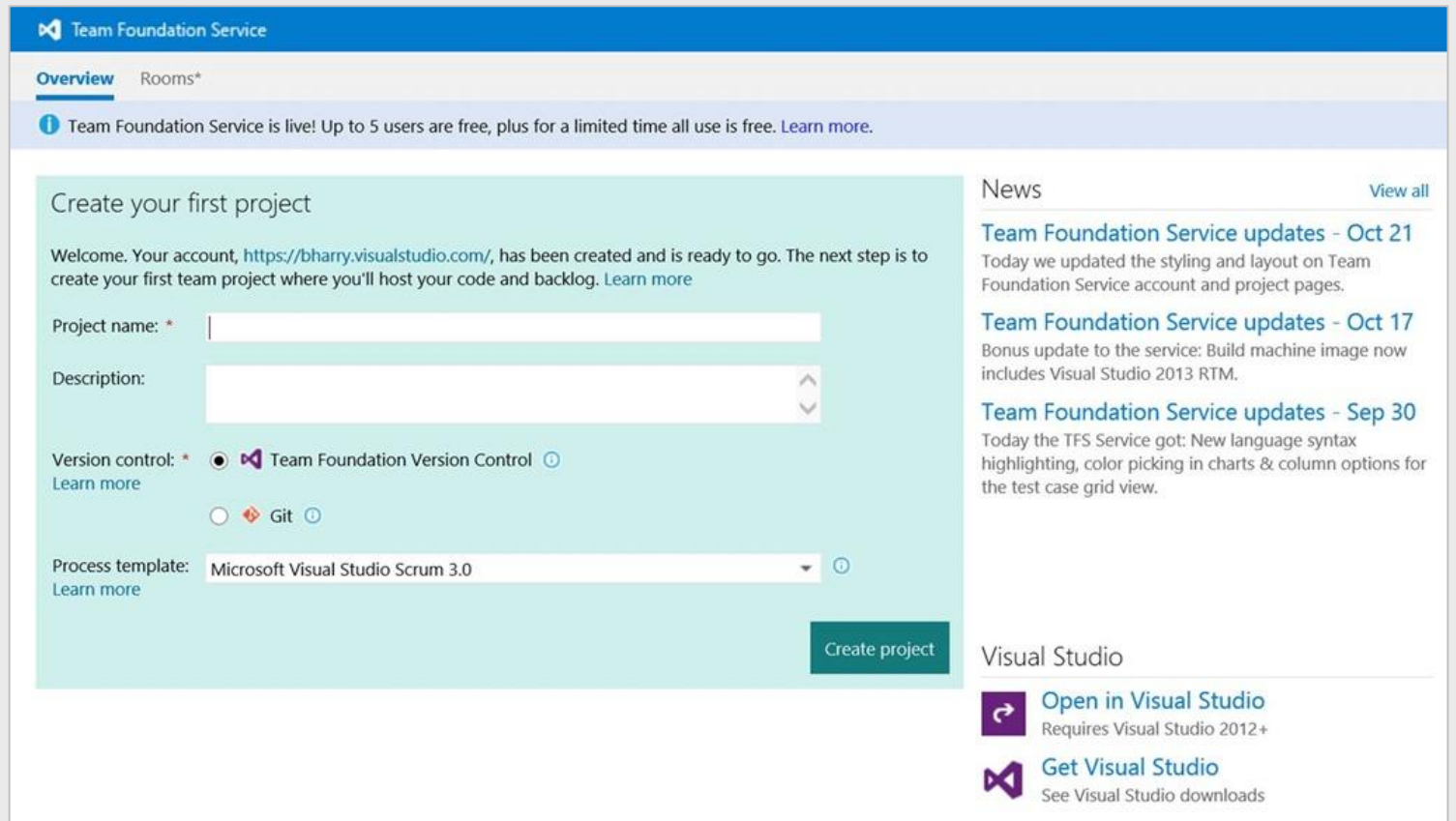
LEARN

We decided we could do better, and made a few changes

Build, measure, learn

BUILD

So we did this...



The screenshot shows the 'Create your first project' page in the Team Foundation Service interface. The page includes a header for 'Team Foundation Service', a navigation bar with 'Overview' and 'Rooms*', and a notification banner stating 'Team Foundation Service is live! Up to 5 users are free, plus for a limited time all use is free. Learn more.' The main content area is a light blue box with the heading 'Create your first project'. It contains a welcome message, a 'Project name' input field, a 'Description' text area, 'Version control' options (Team Foundation Version Control and Git), and a 'Process template' dropdown menu set to 'Microsoft Visual Studio Scrum 3.0'. A 'Create project' button is located at the bottom right of this section. To the right of the form is a 'News' section with three update entries: 'Team Foundation Service updates - Oct 21', 'Team Foundation Service updates - Oct 17', and 'Team Foundation Service updates - Sep 30'. At the bottom right, there are links for 'Visual Studio', 'Open in Visual Studio', and 'Get Visual Studio'.

MEASURE

50%

+20% increase

Top customers contacted by reach-out

Jan	Feb	Δ	Company	Champ	Users (30d)	Work items (30d)	Builds (30d)	Changesets (30d)	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

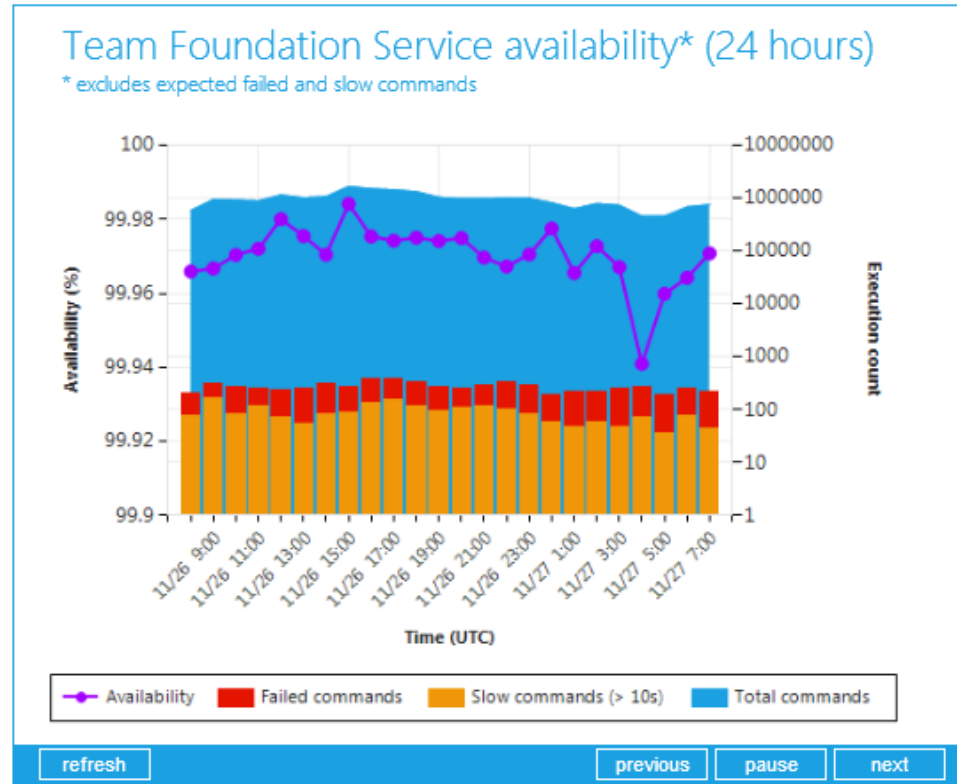
Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Telemetry principles

1. Gather everything
2. Monitor telemetry
3. Root-cause every issue
4. Maintain redundant paths

Team Foundation Service dashboard



Data cached at 11/27/2012 07:20:17 UTC.

Notes

11/19/2012 17:06 UTC

Subsequent availability dip occurred during the config db maintenance portion of today's deployment.

11/19/2012 14:52 UTC

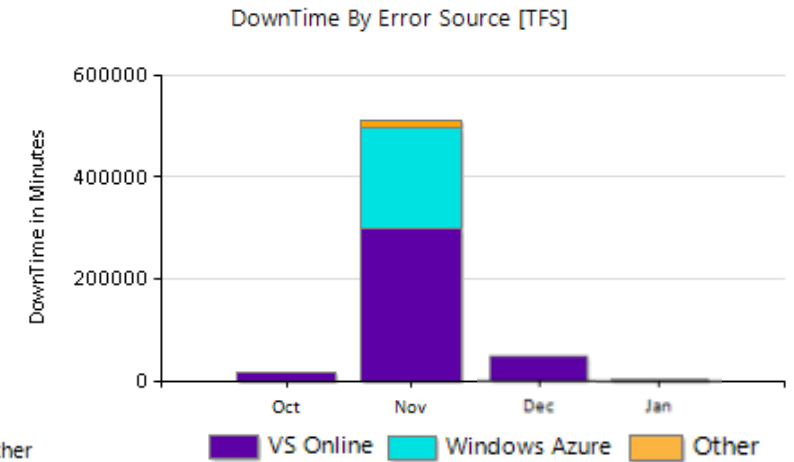
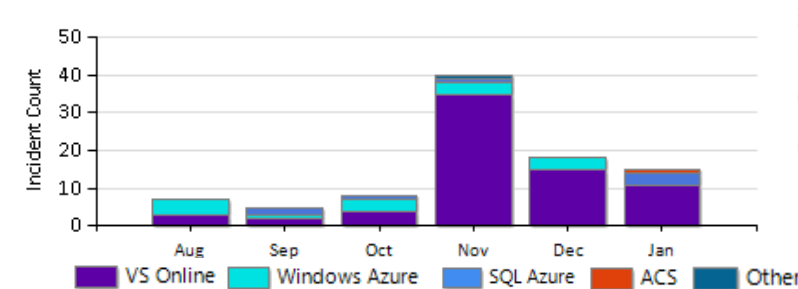
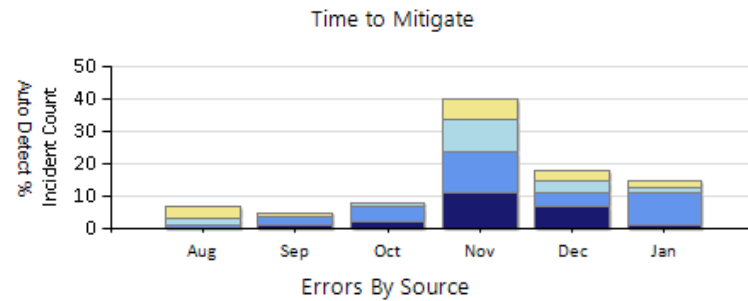
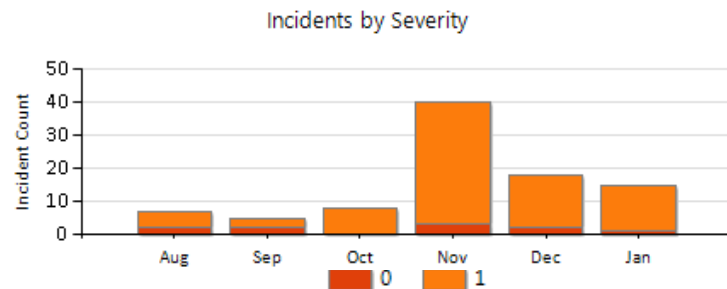
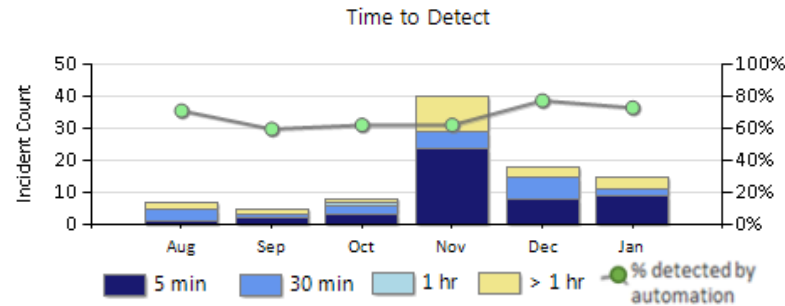
Availability drop is due to the config DB backup copy procedures taking place as part of today's service update pre-requisite deployment activities. Availability is starting to return to normal now that the backup is near

Service availability and operational metrics

Visual Studio Online	Target	Aug '13	Sept'13	Oct'13	Nov'13	Dec'13	Jan'14	Trend	Notes
Service Health - Availability									
Team Foundation Service (99.90%)	0.9990	99.98%	99.98%	99.65%	98.93%	99.60%	99.65%		Using new availability model; Old model = 99.97%
Hosted Build Service (99.90%)	0.9990	99.49%	99.47%	99.72%	98.97%	99.76%	99.90%		
Elastic Load Service (99.90%)	0.9990	99.88%	99.97%	99.99%	99.96%	99.96%	100.00%		
App Insights (99.50%)	0.9950	99.50%	99.89%	97.40%	92.40%	99.20%	99.00%		1/18 & 1/22 VS Portal
GSM (99.50%)	0.9950	97.60%	99.99%	99.98%	99.07%	98.10%	97.40%		1/18 & 19 SQL Azure connectivity 1/20 ACS outage
Monaco	0.9950								TBD

Customer Impacting Incidents (Sev 0-1)								
Team Foundation Service		2	2	1	9	3	2	
Hosted Build Service		4	1	3	1	0	0	
Elastic Load Service		1	2	2	3	1	0	
App Insights		0	0	2	29	14	8	
GSM		1	0	0	1	2	5	
Monaco							0	
TOTALS:		8	5	8	43	20	15	

- TFS availability was impacted by S59 deployment
- Moving to regular deployment schedule from S61
- AI LSI trend is improving
- Added Monaco
- Platform issues stabilizing



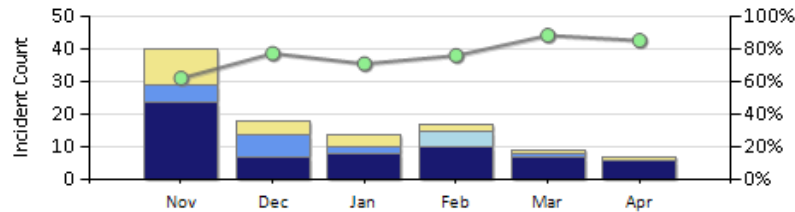
Service availability and operational metrics

Visual Studio Online	Target	Nov'13	Dec'13	Jan'14	Feb'14	Mar'14	Apr'14	Trend	Notes
Service Health - Availability									
Team Foundation Service (99.90%)	99.9%	98.93%	99.60%	99.65%	99.78%	99.69%	99.73%		Old model = 99.99%
Hosted Build Service (99.90%)	99.9%	98.97%	99.76%	99.90%	99.83%	99.81%	99.98%		
Elastic Load Service (99.90%)	99.9%	99.96%	99.96%	100.00%	100.00%	99.97%	100.00%		
App Insights (99.50%)	99.5%	92.40%	99.20%	99.00%	98.80%	99.10%	99.65%		
GSM (99.50%)	99.5%	99.07%	98.10%	97.40%	98.70%	99.10%	99.60%		
Monaco	99.5%	97.85%	97.62%	98.20%	98.84%	99.41%	99.00%		

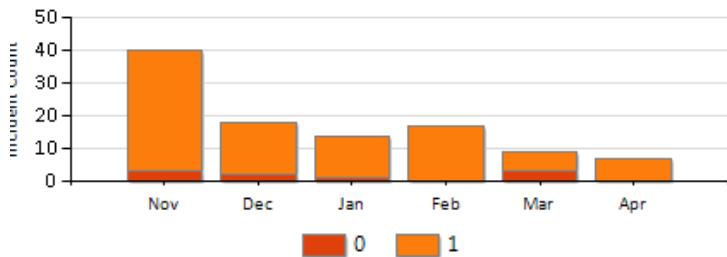
Customer Impacting Incidents (Sev 0-1)								
Team Foundation Service		9	3	2	2	1	1	
Hosted Build Service		1	0	0	2	0	1	
Elastic Load Service		3	1	0	3	2	0	
App Insights		29	14	8	10	6	5	
GSM		1	2	5	0	0	0	
Monaco				0	0	0	0	
TOTALS:		43	20	15	17	9	7	

- Overall decline in month-over-month incidents
- Uneventful //build launch
- App Insights Sev0/1 declining but Sev2 have increased

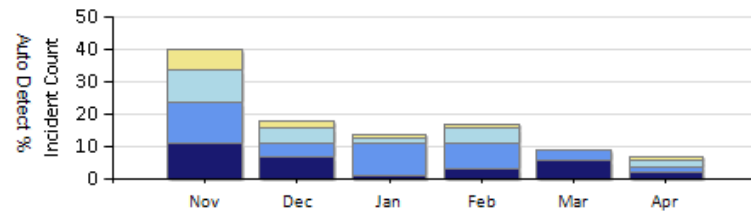
Time to Detect



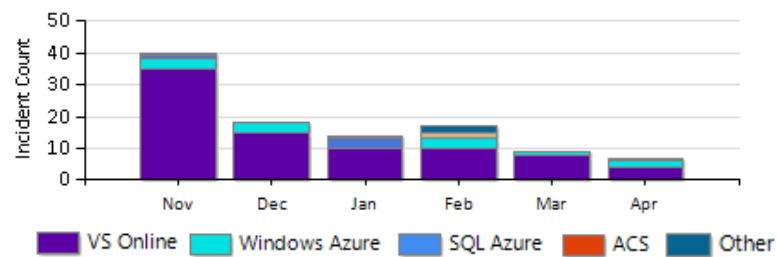
Incidents by Severity



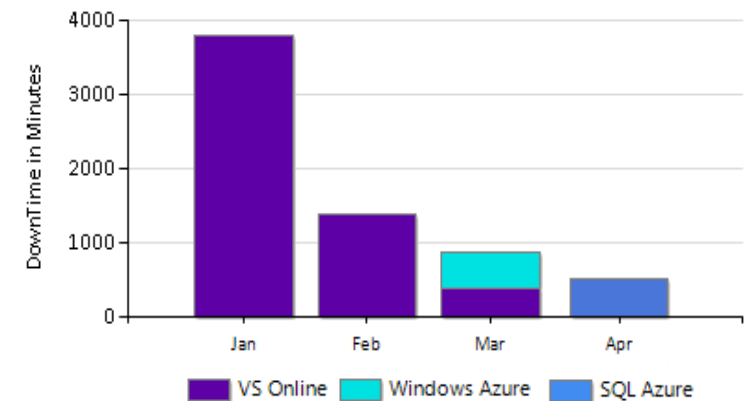
Time to Mitigate



Errors By Source



DownTime By Error Source [TFS]

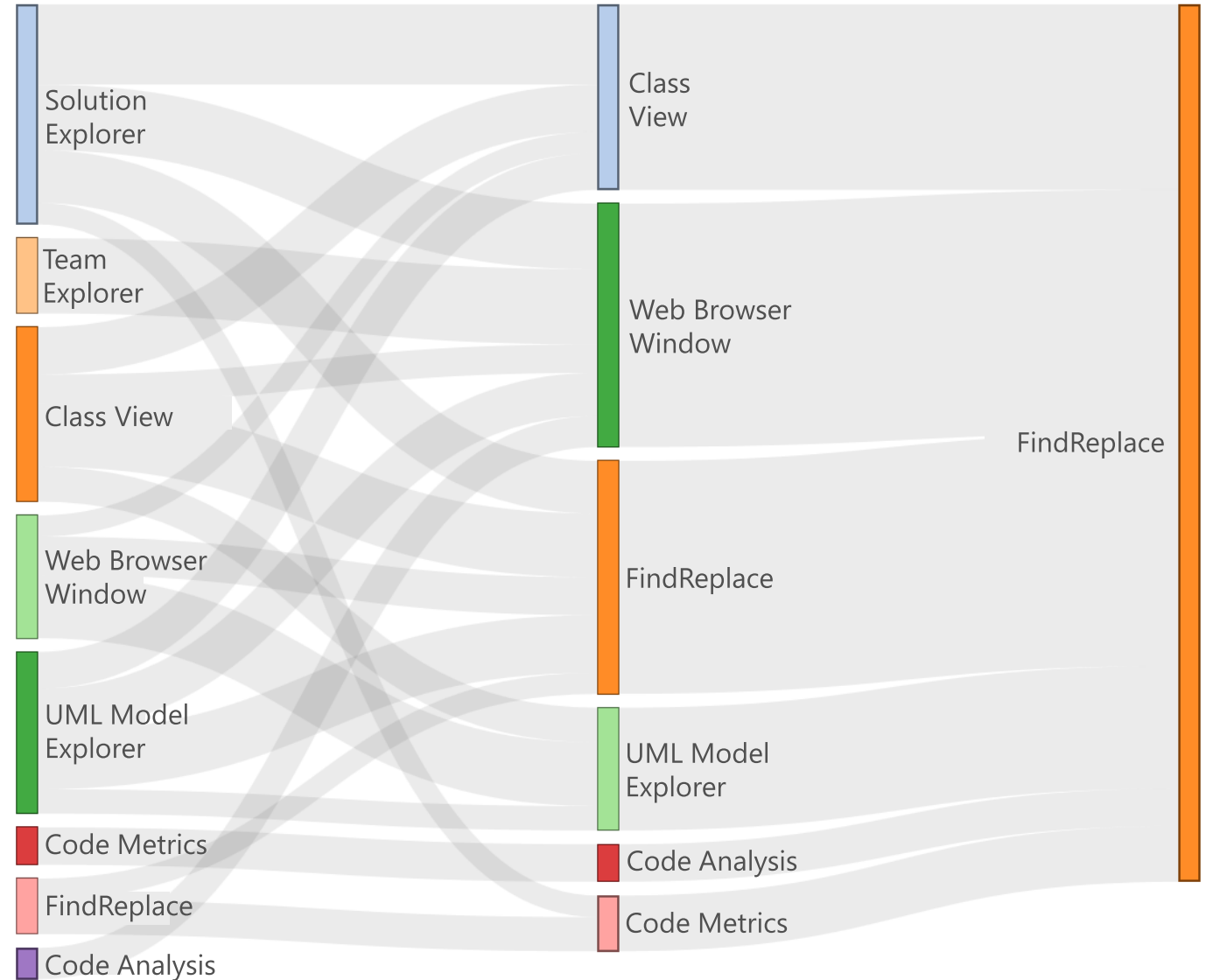


Dashboard mockups: usage patterns

Based on Tools window activations

Easy to combine with command data

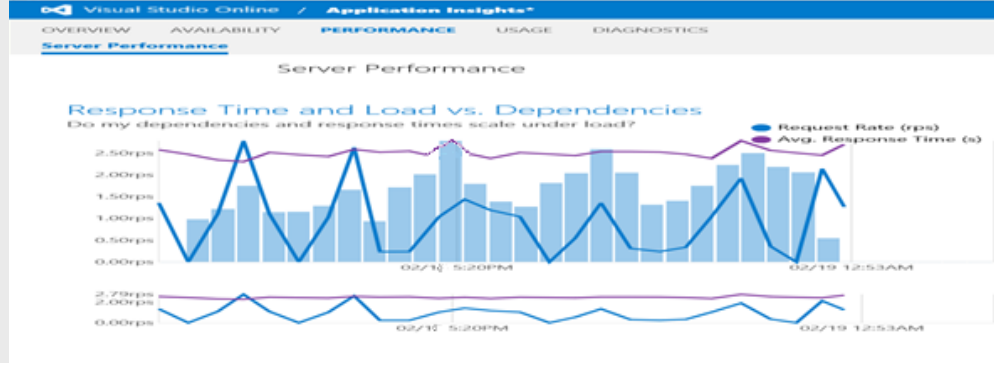
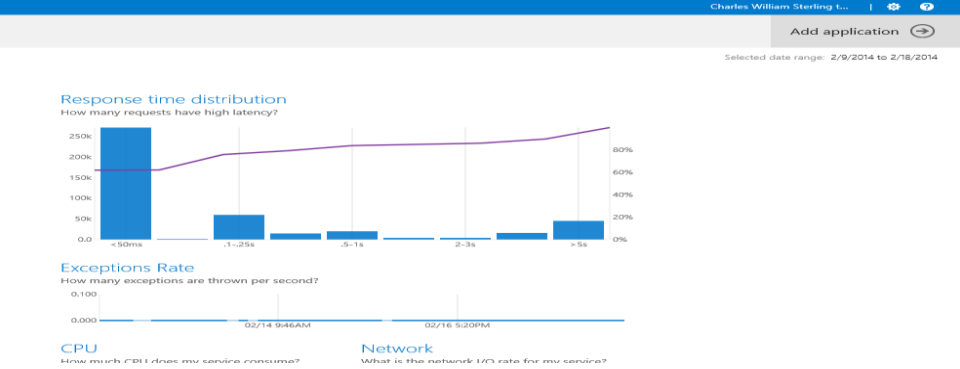
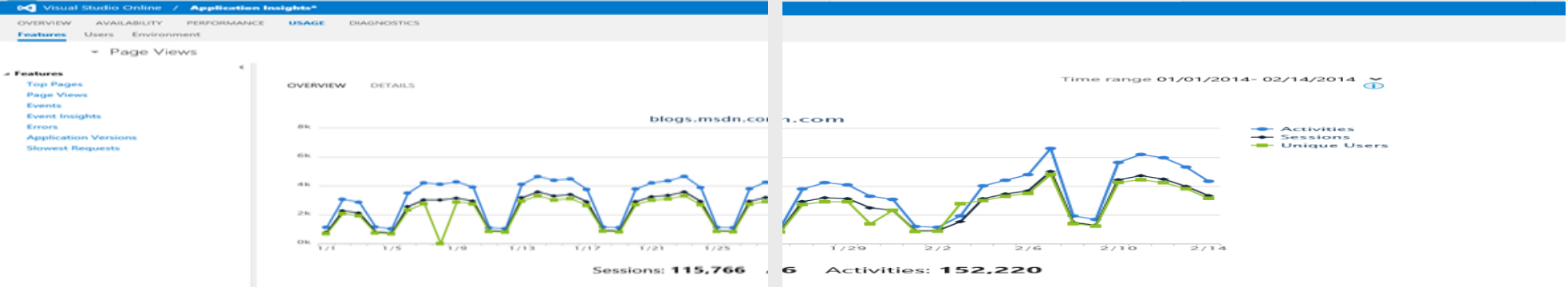
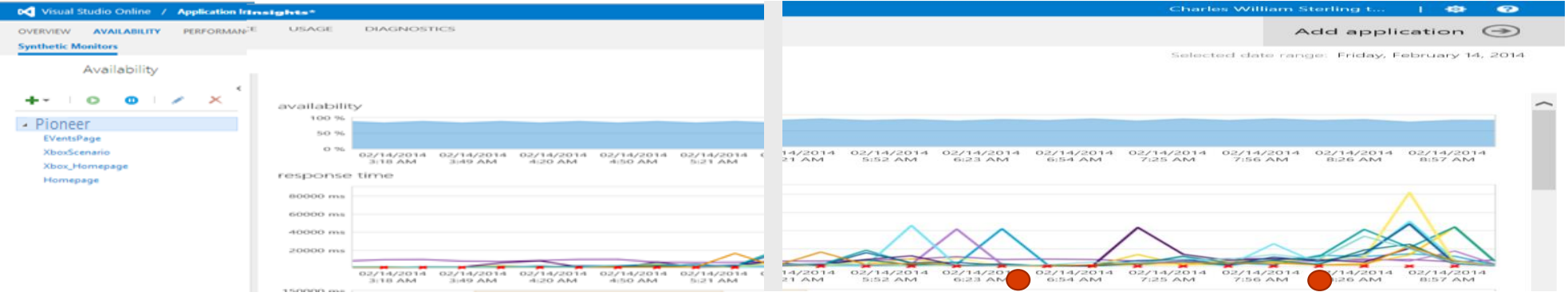
Can be broken out/filtered by any of the common slicers (SKU, release, language, OS, etc.)



Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Application insights



Adopting DevOps

The why?	Teams	Schedule ceremonies	Deployment practices
Live site culture	Business success	Telemetry learning	Productizing

Dimensions of practice

Are these the dimensions you see? How do you assess improvement?

Public/hybrid
cloud

Agile
schedule
and team

Managed
technical
debt

Production
first

Flow of value

Evidence
and data

Hypothesis-
based backlog



THERE'S NO PLACE LIKE PRODUCTION

