



DevOps Transformation In Microsoft

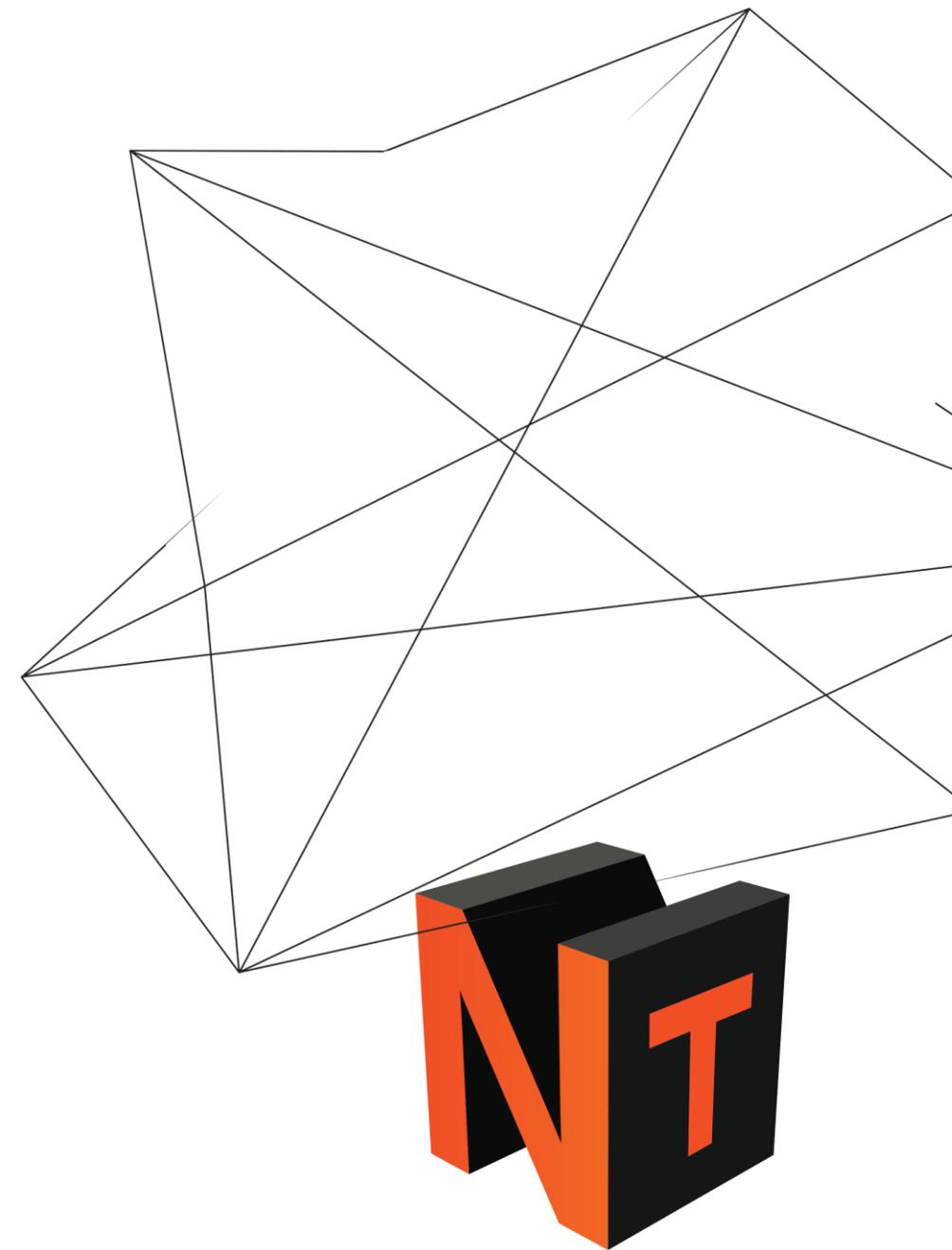
Ognjen Bajić

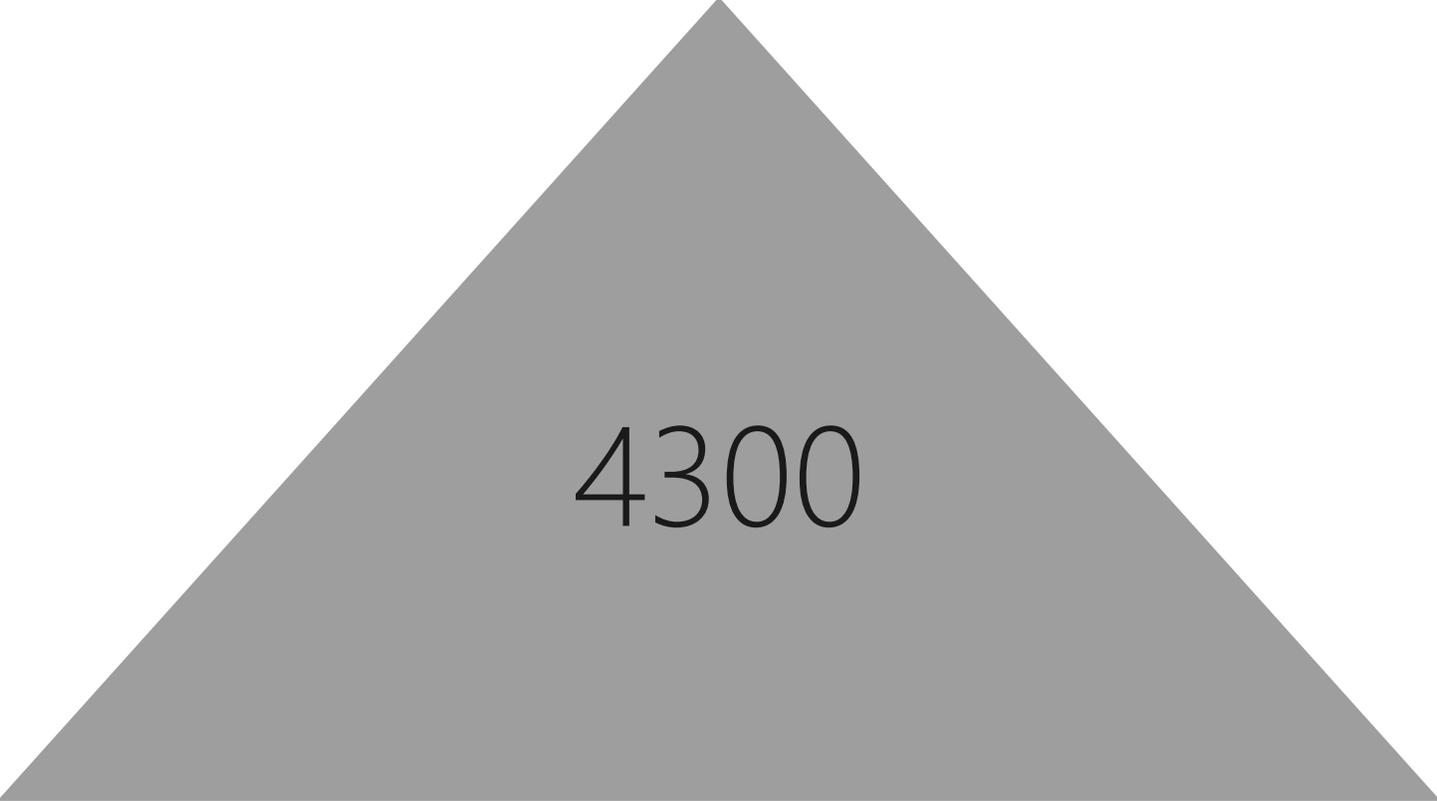
Ana Roje Ivančić

MVPs for Visual Studio and Development Technologies

Ekobit, Croatia

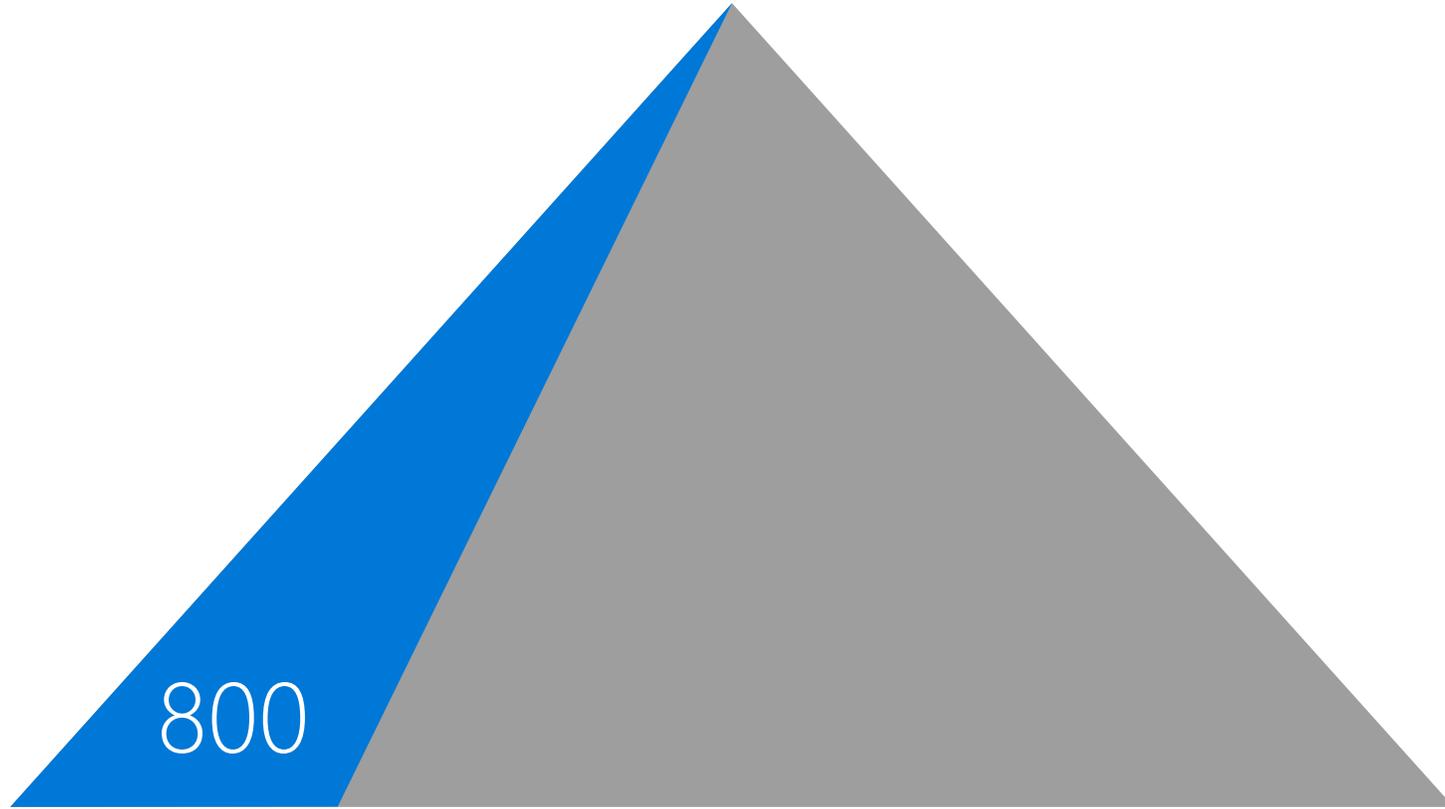
#ntk18



A gray triangle pointing upwards, centered on the page. Inside the triangle, the number 4300 is written in a black, sans-serif font.

4300

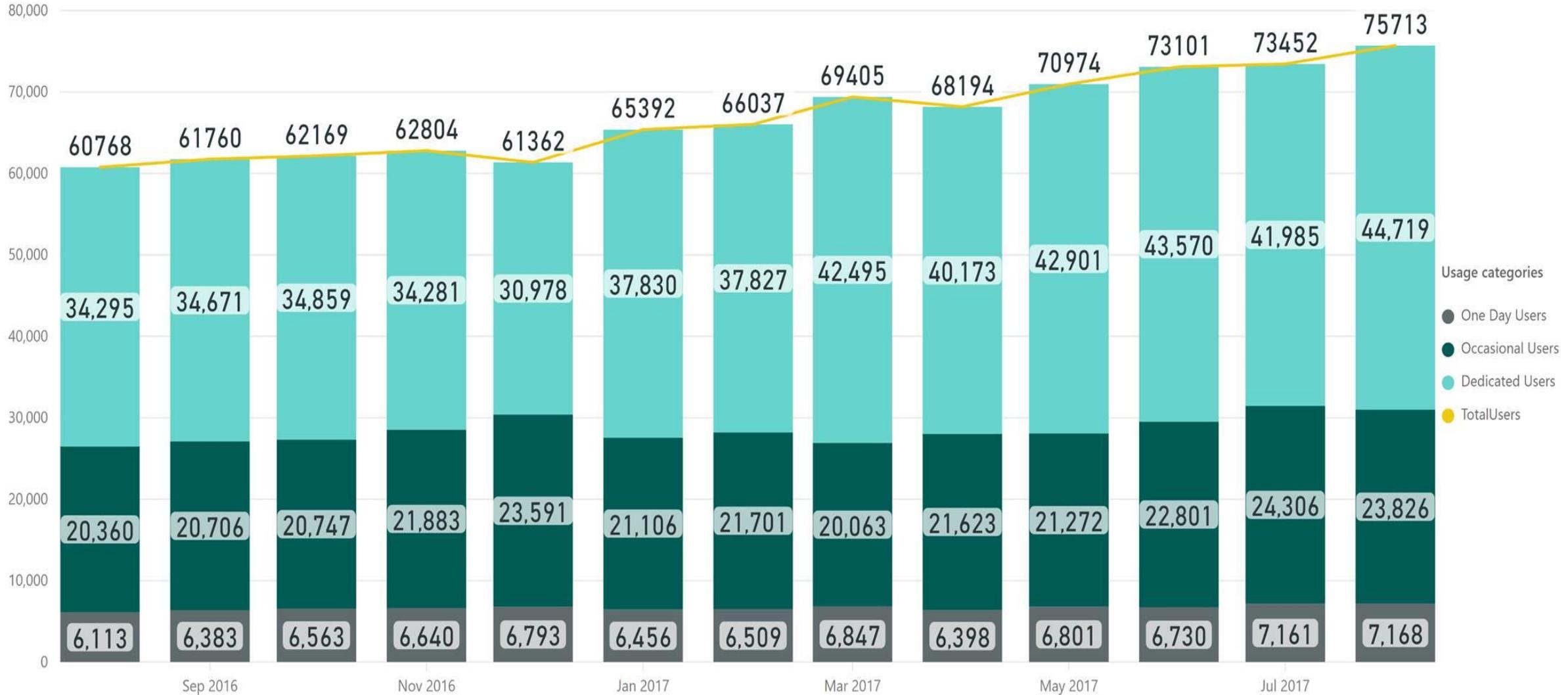
The Developer Division at Microsoft



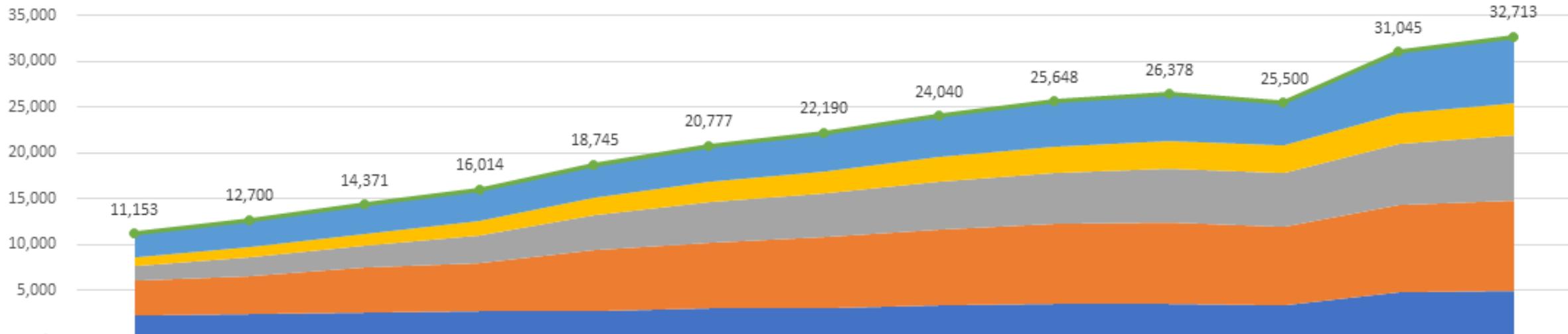
The VSTS team... spread out across 60+ feature teams

Microsoft uses VSTS – One Engineering System (1ES)

Companywide VSTS Users of WIT, Git, and/or Build



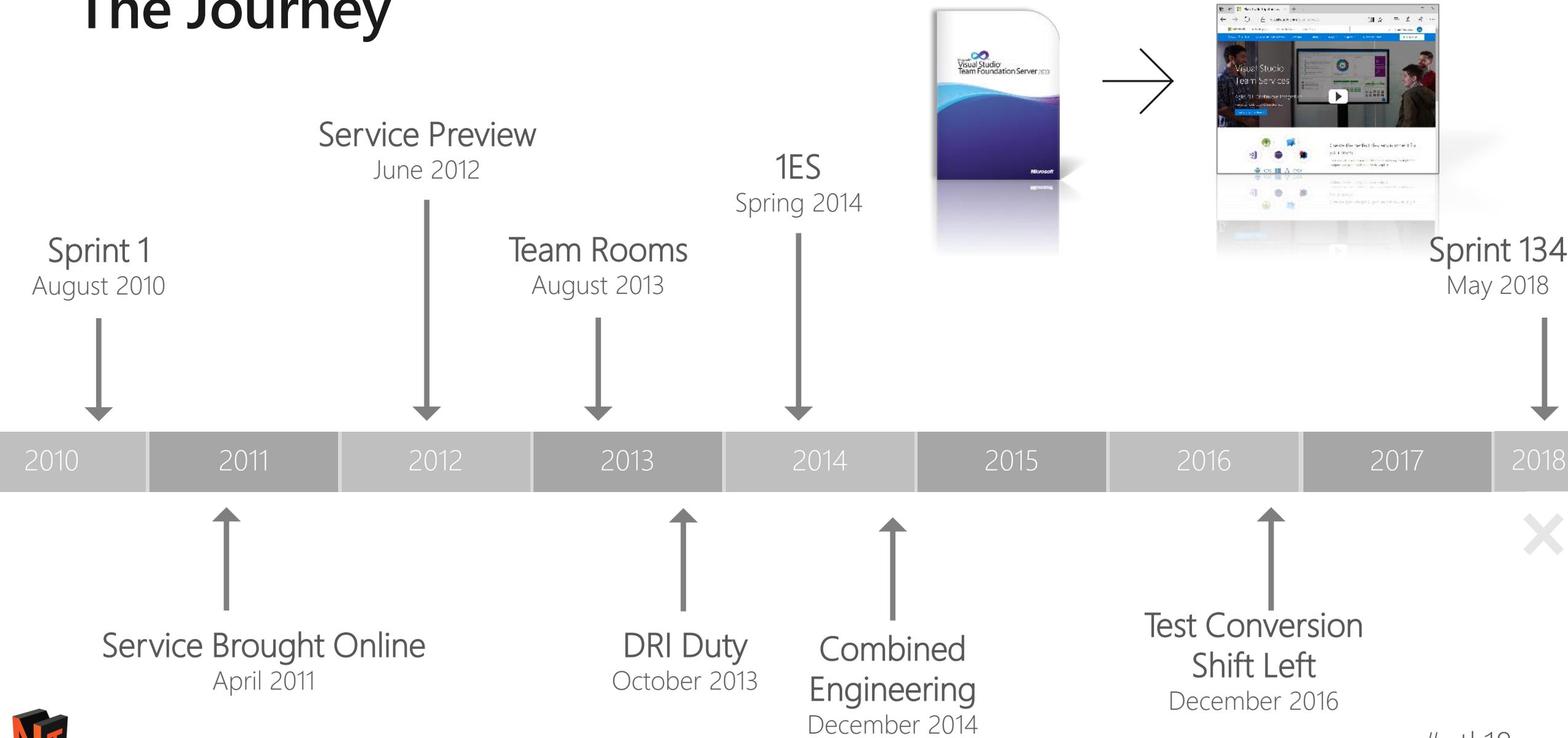
VSTS: >30k Microsoft wide Git users



	February 2016	March 2016	April 2016	May 2016	June 2016	July 2016	August 2016	September 2016	October 2016	November 2016	December 2016	January 2017	February 2017
WDG	2,523	2,914	3,157	3,382	3,654	3,957	4,168	4,568	4,947	5,048	4,685	6,797	7,255
Strategy & Ops	945	1,123	1,361	1,684	1,949	2,257	2,449	2,669	2,907	3,122	3,043	3,328	3,507
Other	1,696	2,070	2,423	2,936	3,692	4,376	4,765	5,204	5,544	5,879	5,900	6,635	7,120
C&E	3,674	4,212	4,847	5,347	6,709	7,206	7,762	8,317	8,671	8,773	8,506	9,498	9,921
AIR & OPG	2,315	2,381	2,583	2,665	2,741	2,981	3,046	3,282	3,579	3,556	3,366	4,787	4,910
Grand Total	11,153	12,700	14,371	16,014	18,745	20,777	22,190	24,040	25,648	26,378	25,500	31,045	32,713



The Journey



#ntk18

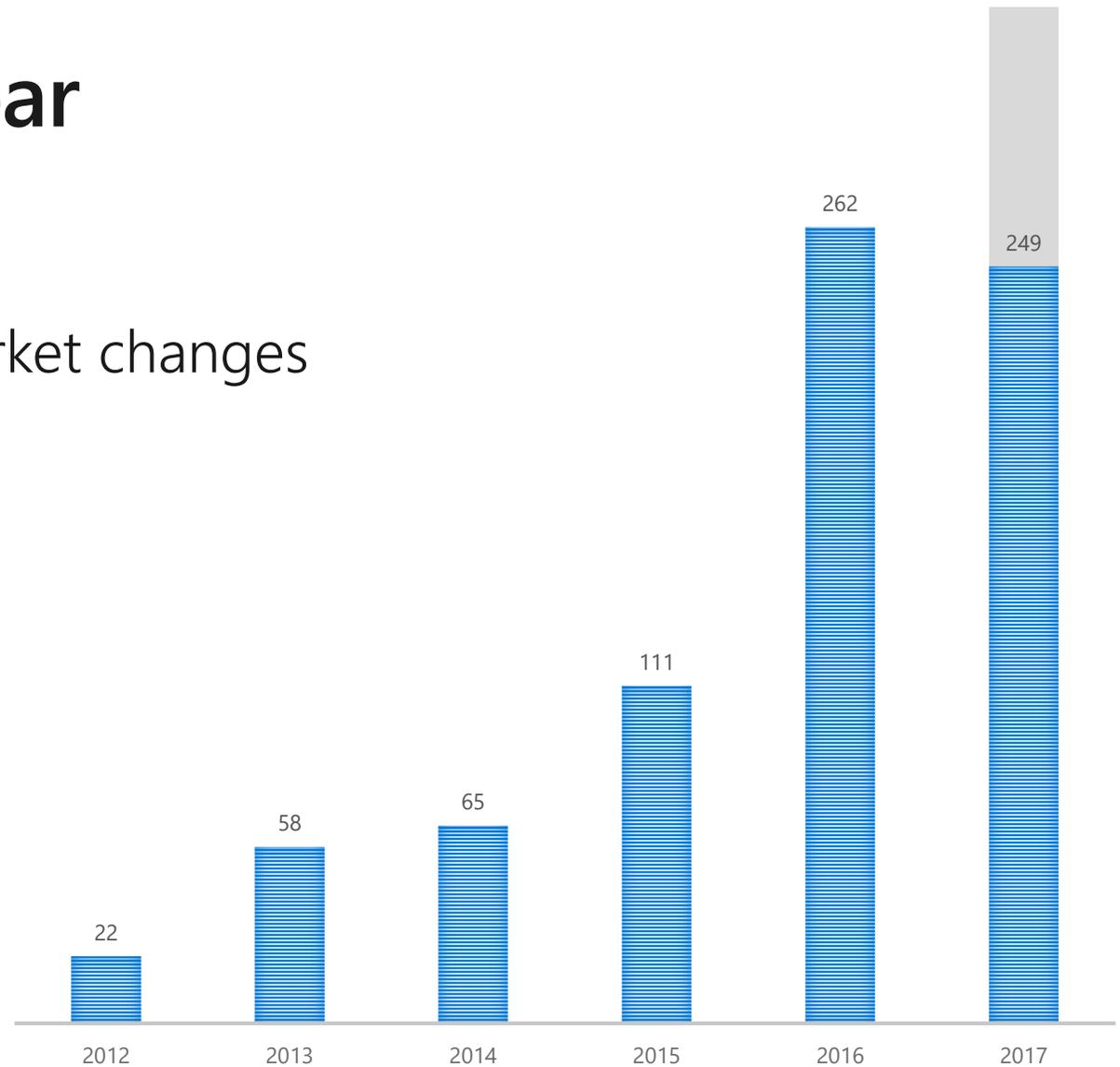
Features Delivered per Year

Deliver more value to customers

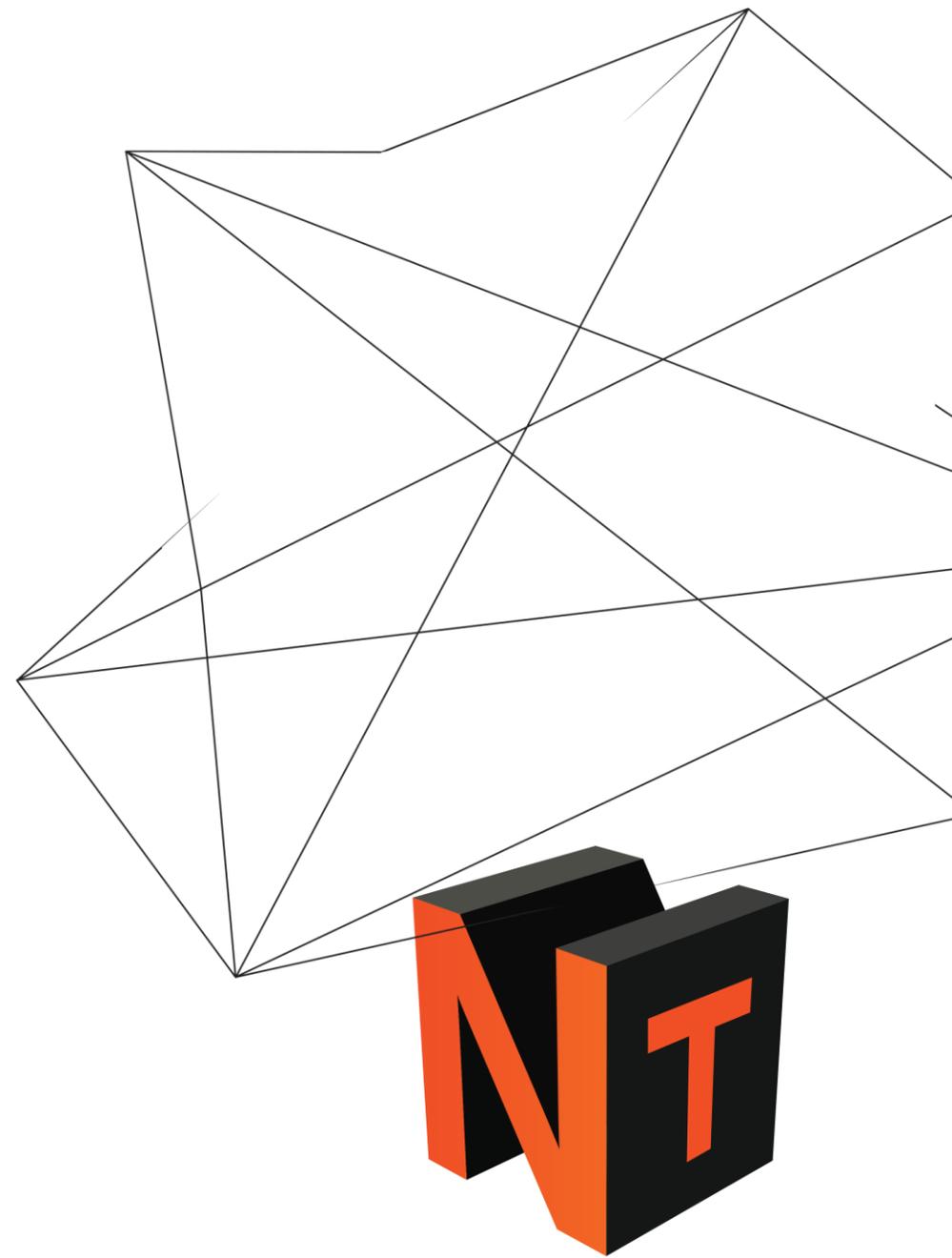
Faster responses to customers and market changes

Improved engineering satisfaction

2x productivity increase



What it looked like before...



#ntk18

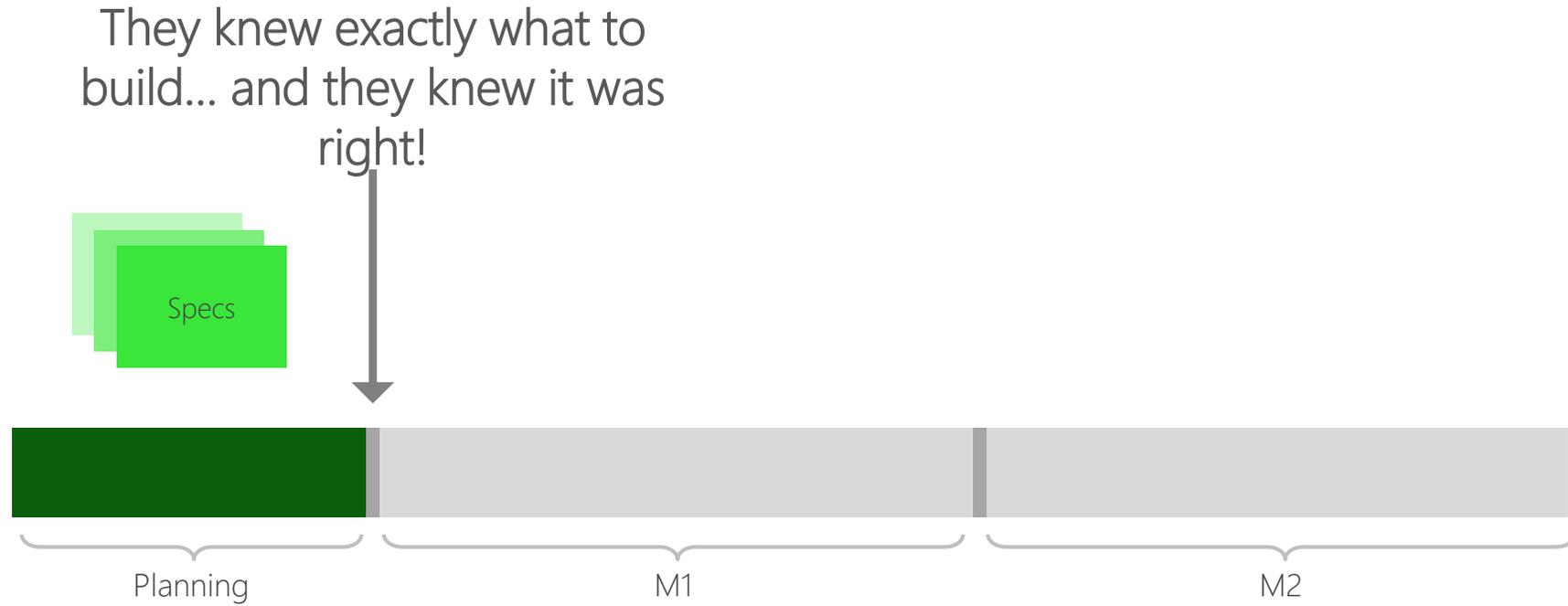
Before

The OLD way



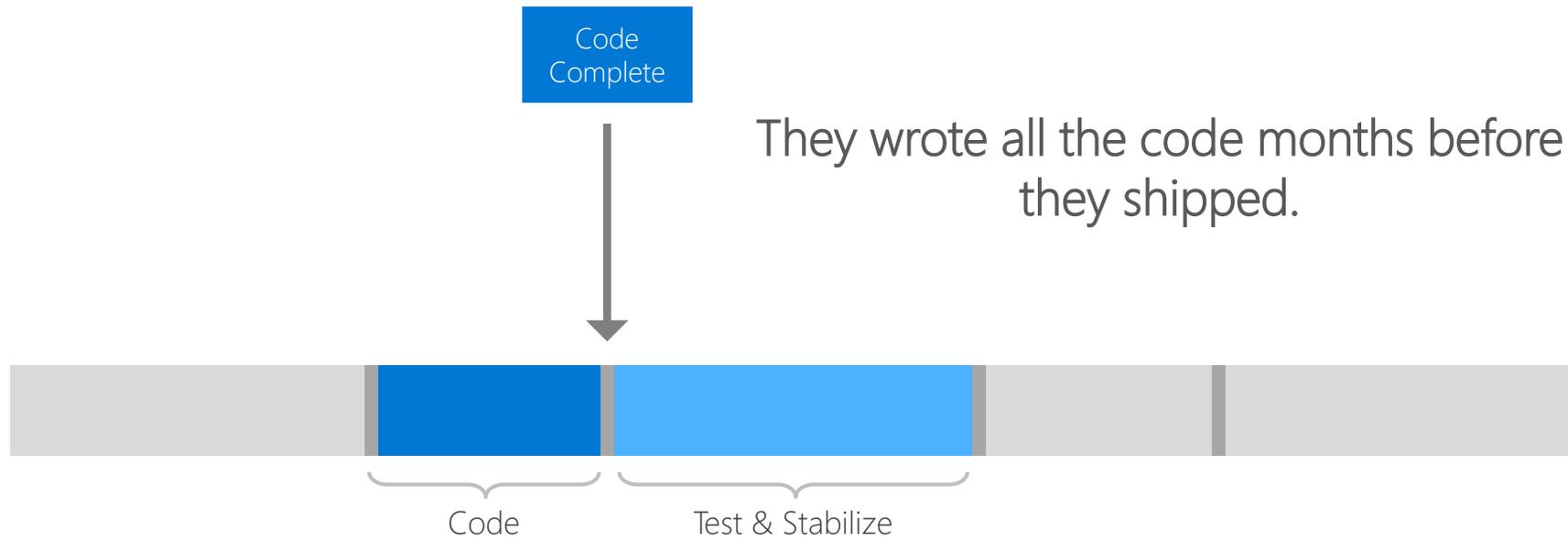
Before

The OLD way



Before

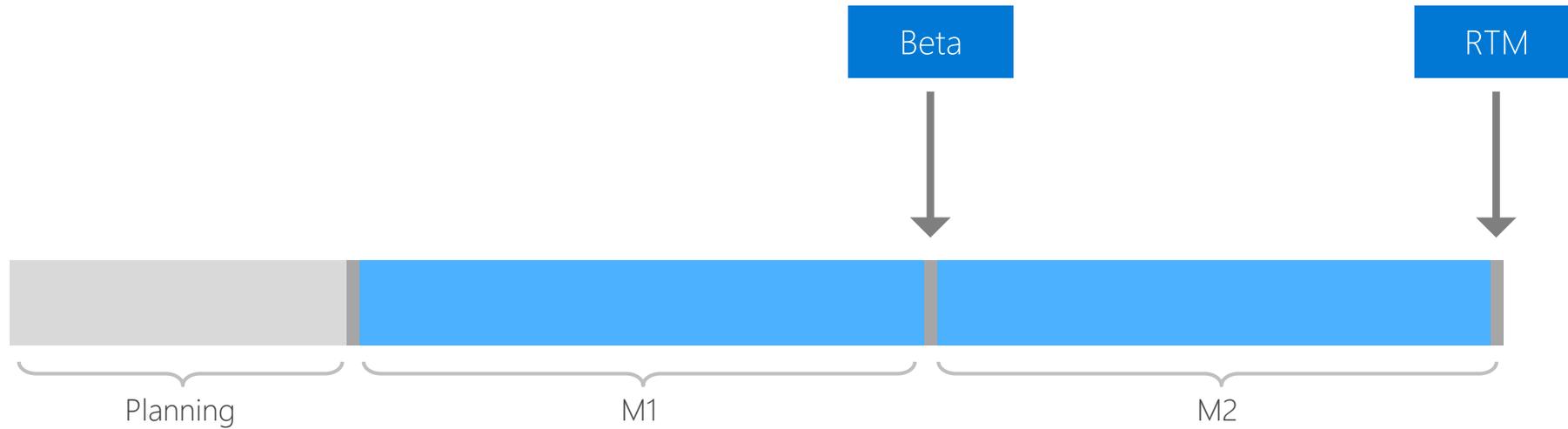
The OLD way



Before

The OLD way

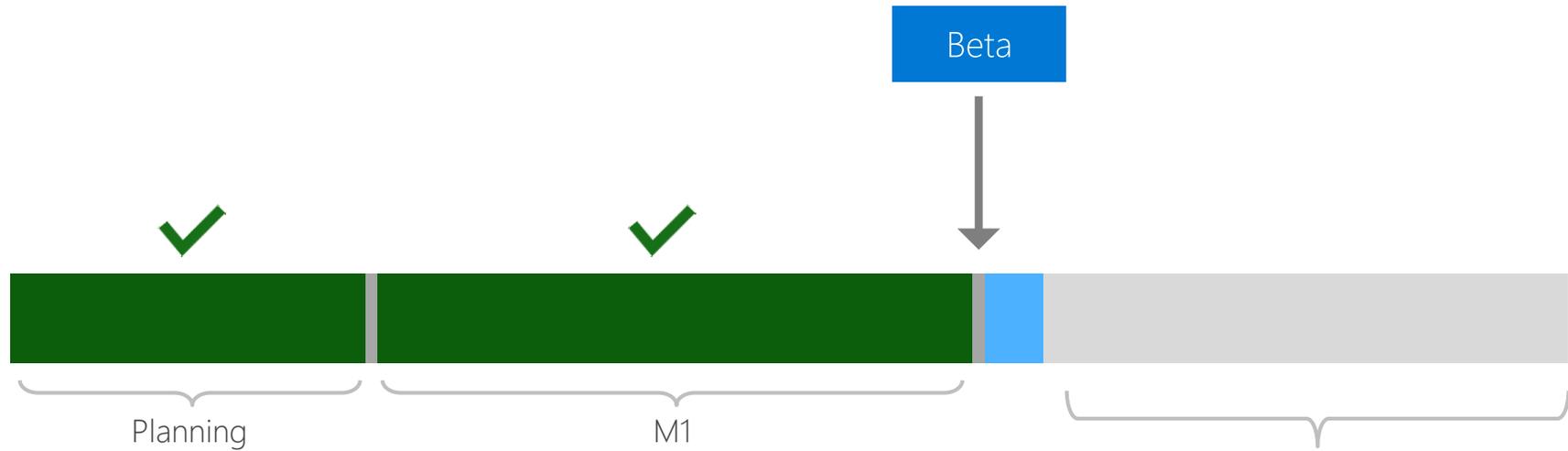
They had a perfect schedule and knew exactly when it would be ready!



Before

The OLD way

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



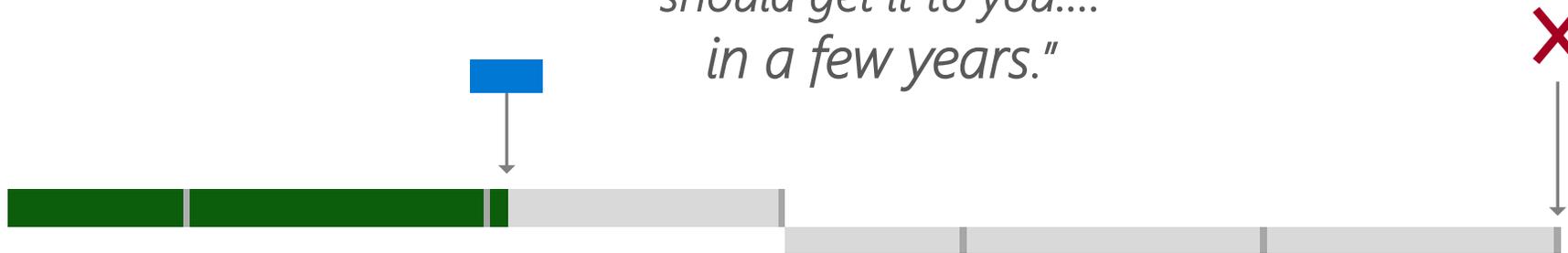
... but they were booked solid already.



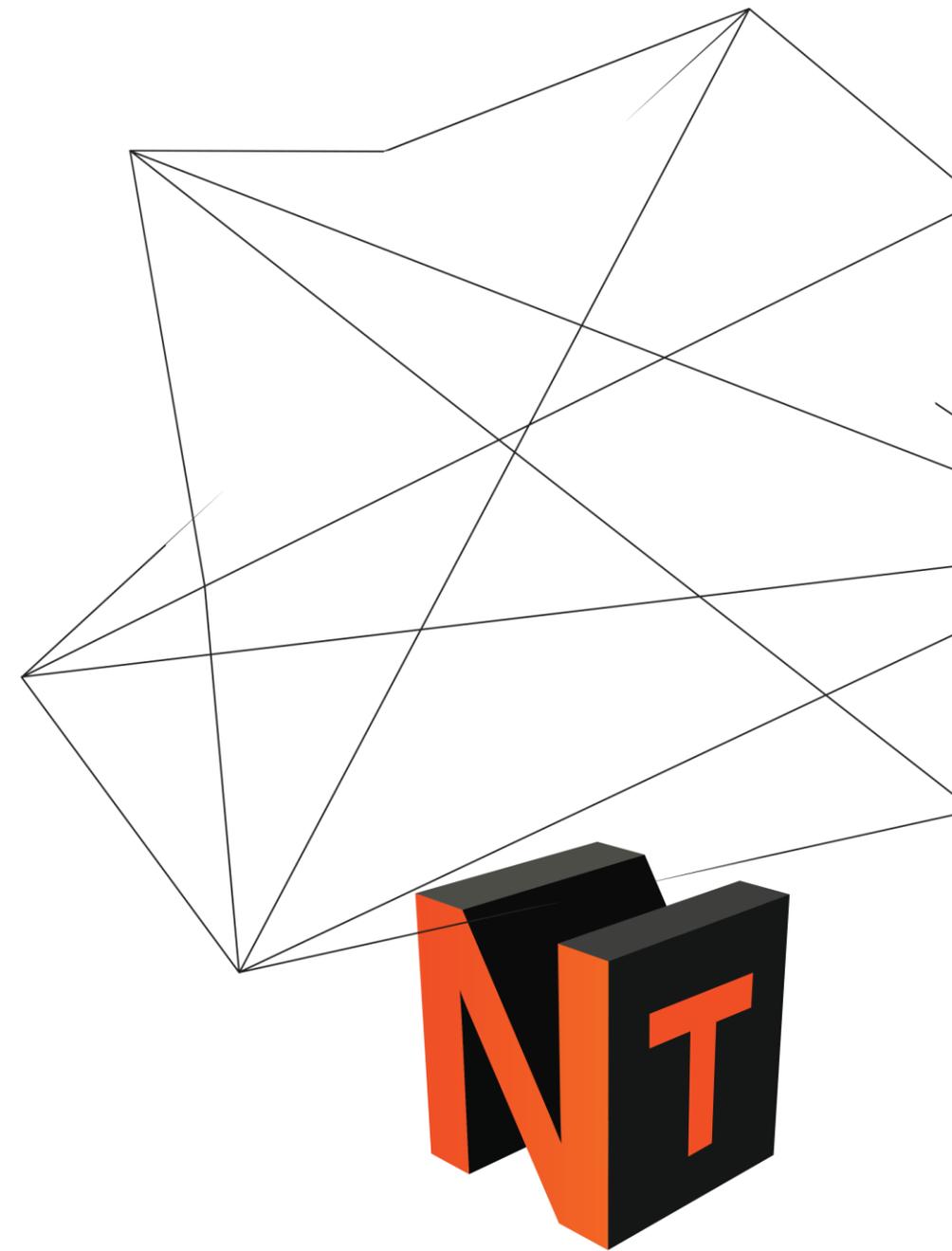
Before

The OLD way

"Great feedback. Thanks! We'll take a look in planning for the next release. We should get it to you... in a few years."

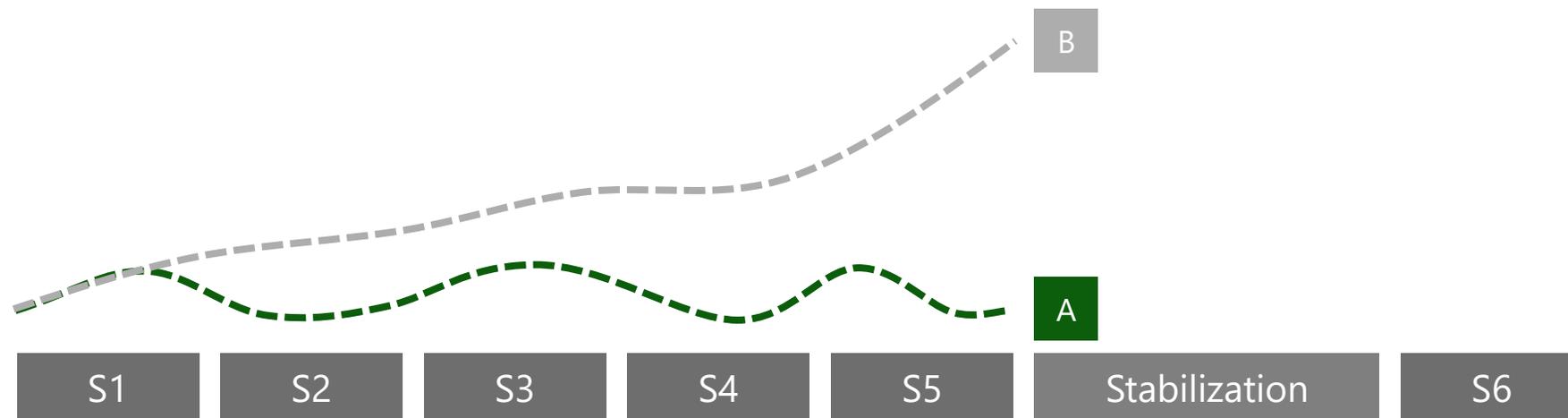


A: Very well in the era in which it was born. But...



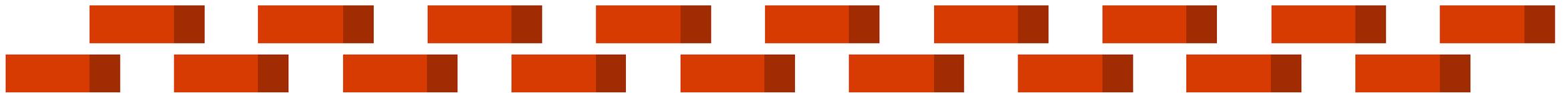
Story: Sprint 1-5 and the “Stabilization” Sprint

“Let’s do this Agile thing... but we should probably reserve some time to stabilize things.”



Now

3 weeks

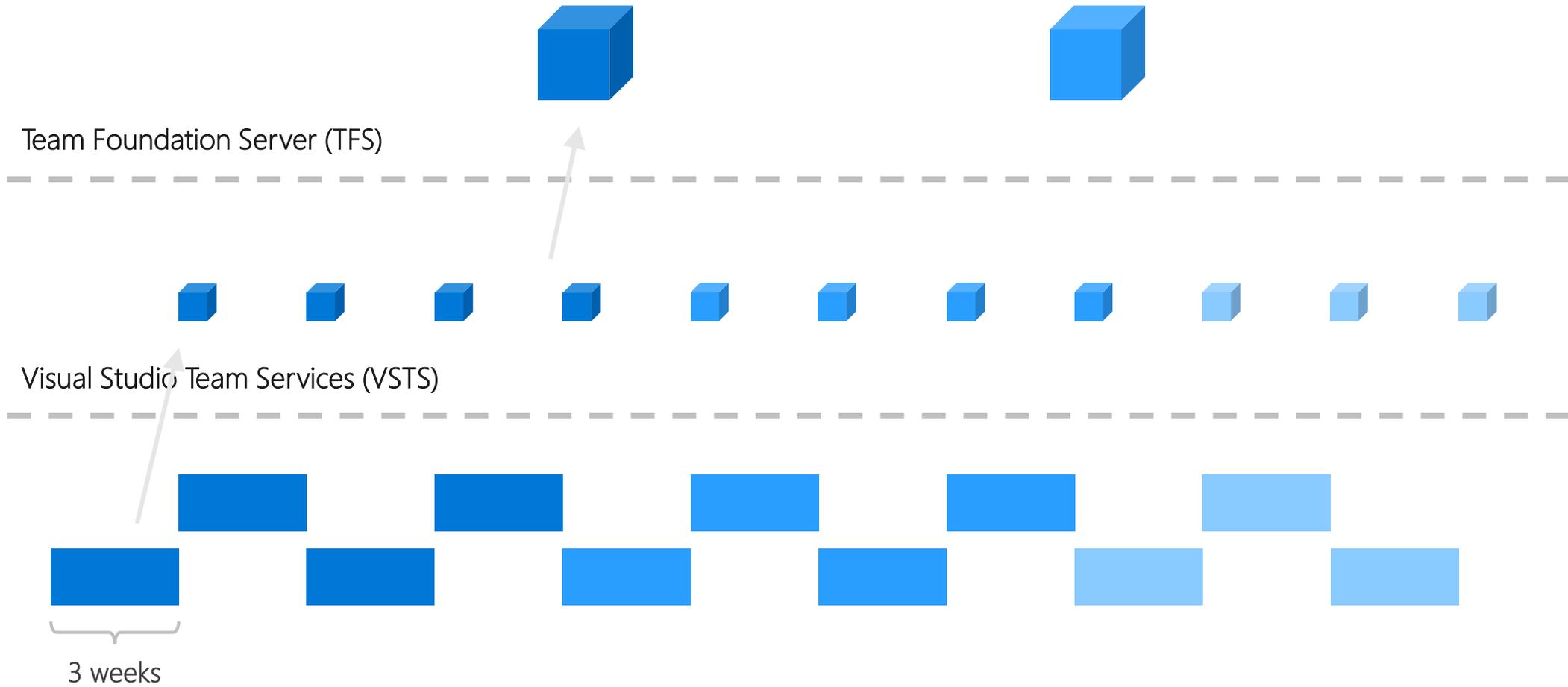


2 years

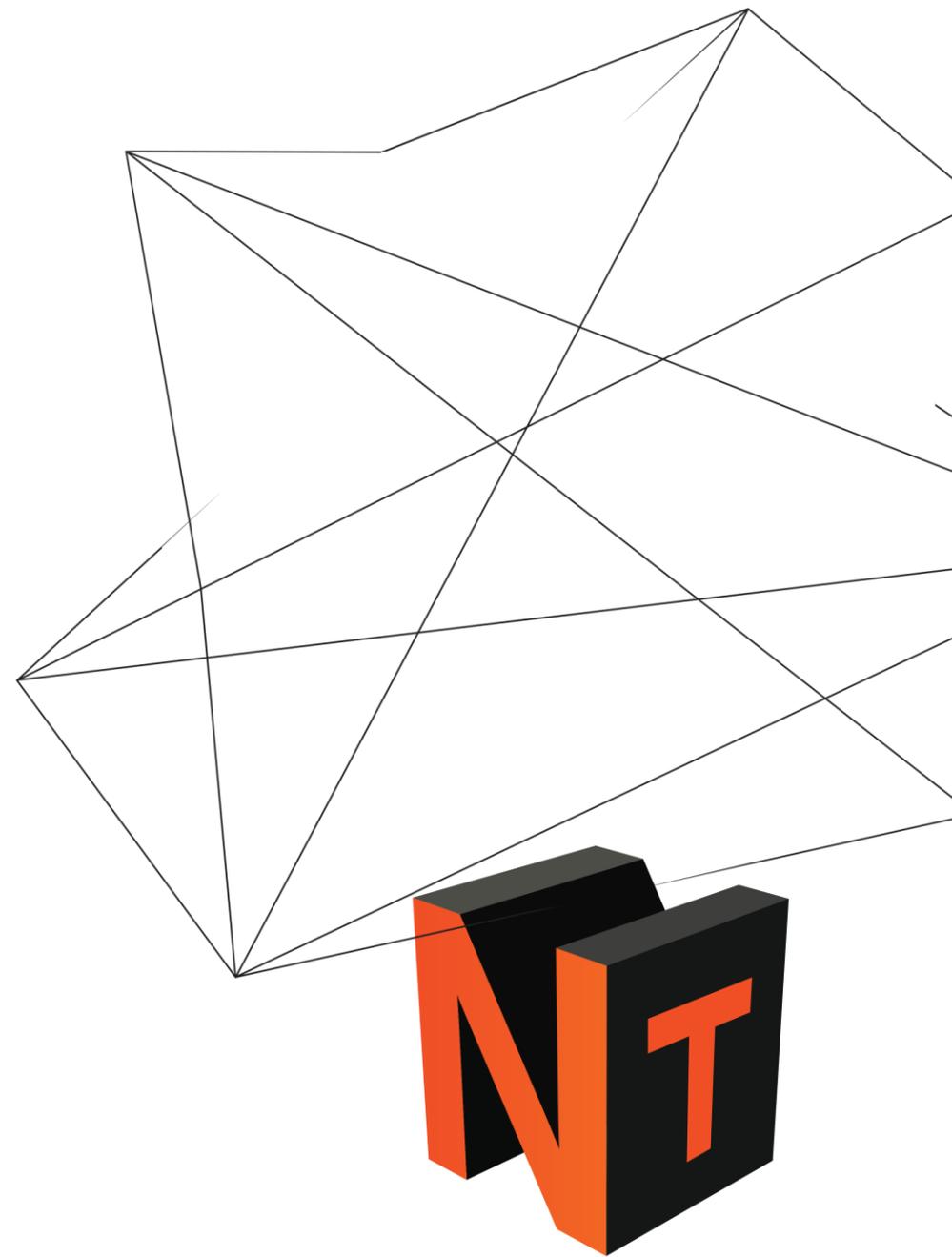


#ntk18

How do they work now?



Results?



#ntk18

Transformation

Before

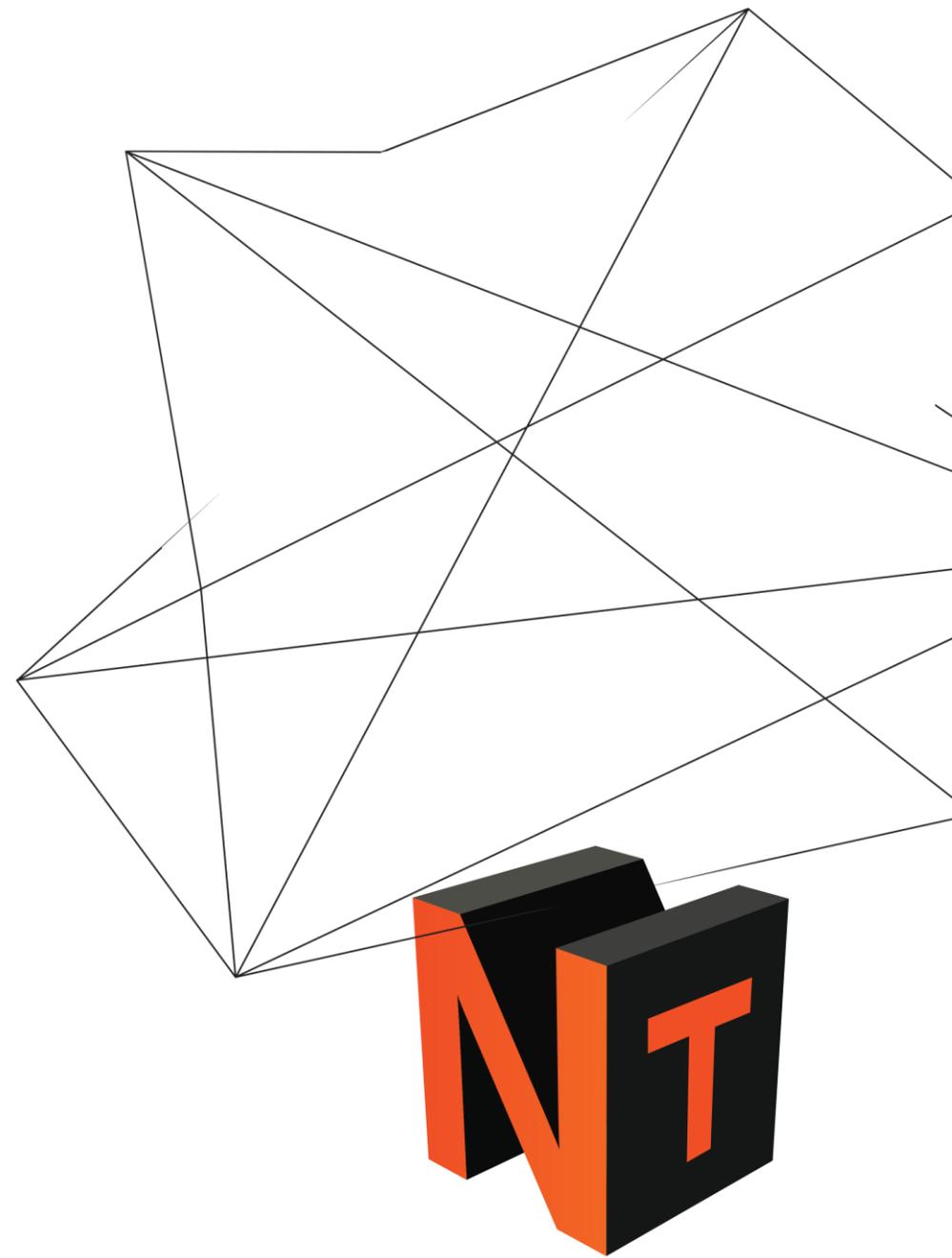
- 4-6 month milestones
- Horizontal teams
- Personal offices
- Long planning cycles
- PM, Dev, Test
- Yearly customer engagement
- Feature branches
- 20+ person teams
- Secret roadmap
- Bug debt
- 100 page spec documents
- Private repositories
- Deep organizational hierarchy
- Success is a measure of install numbers
- Features shipped once a year

After

- 3-week sprints
- Vertical teams
- Team rooms
- Continual Planning & Learning
- PM & Engineering
- Continual customer engagement
- Everyone in master
- 8-12 person teams
- Publicly shared roadmap
- Zero debt
- Specs in PPT
- Open source
- Flattened organization hierarchy
- User satisfaction determines success
- Features shipped every sprint



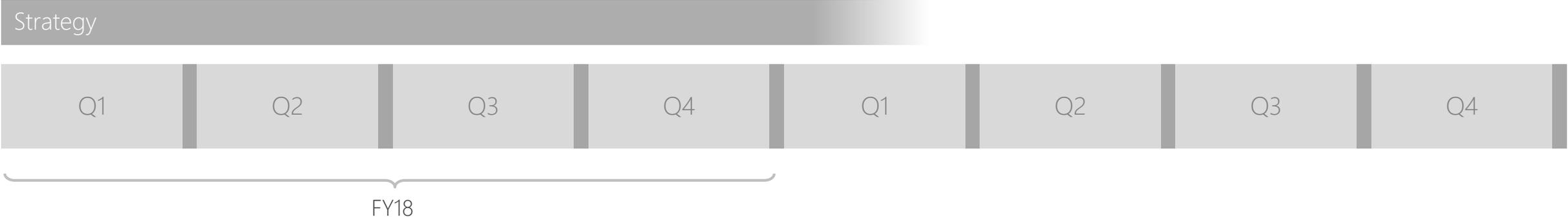
**Plans are worthless,
but planning is everything.**



Planning



Planning



Planning

6 month plan

Strategy

Q1

Q2

Q3

Q4

Q1

Q2

Q3

Q4

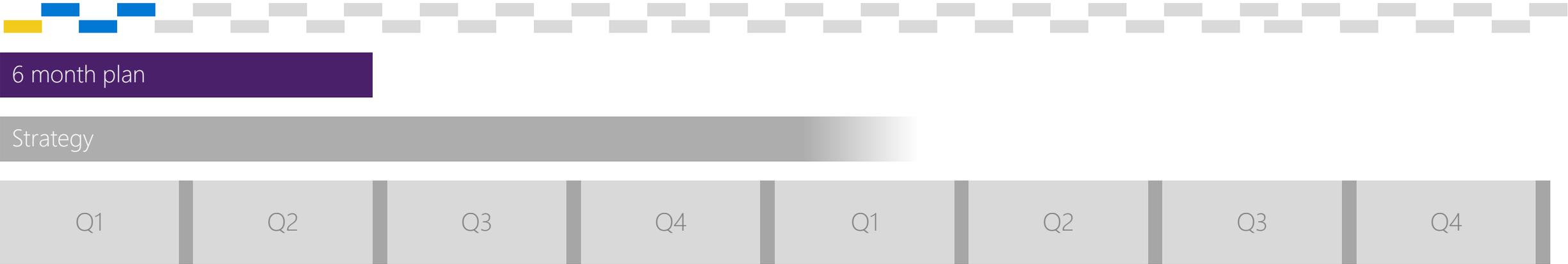
FY18 H1



Planning



Planning



Planning



6 month plan

Strategy



Planning



6 month plan

Strategy



Planning

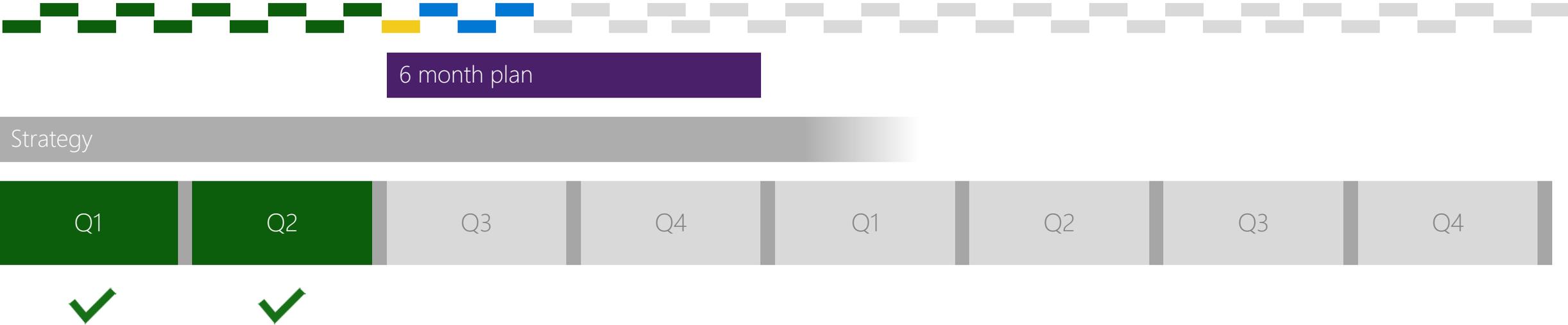


6 month plan

Strategy



Planning

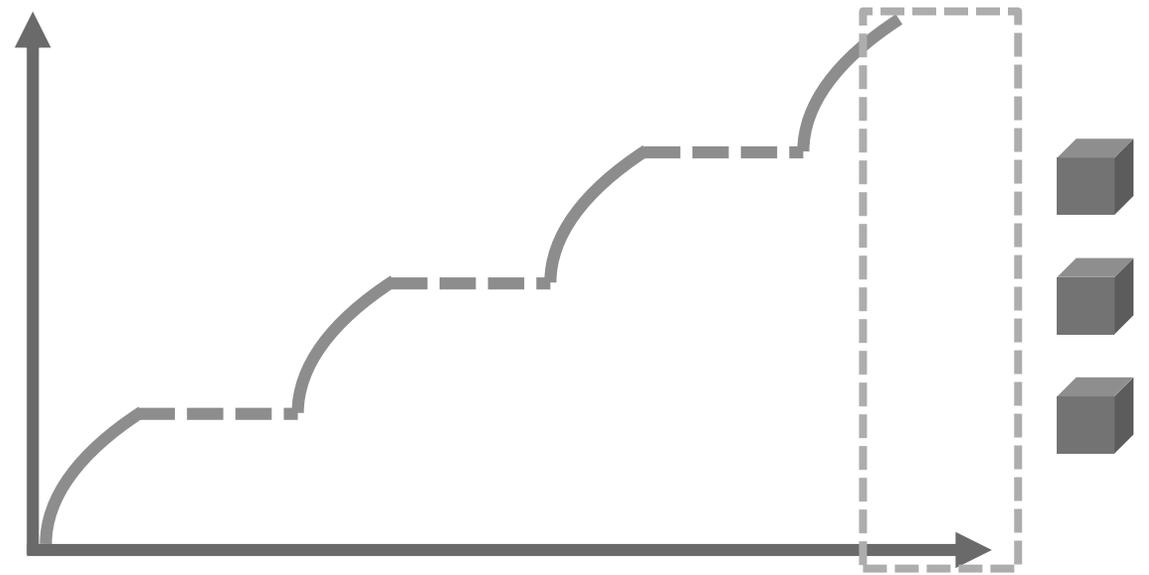


Continual Planning & Learning

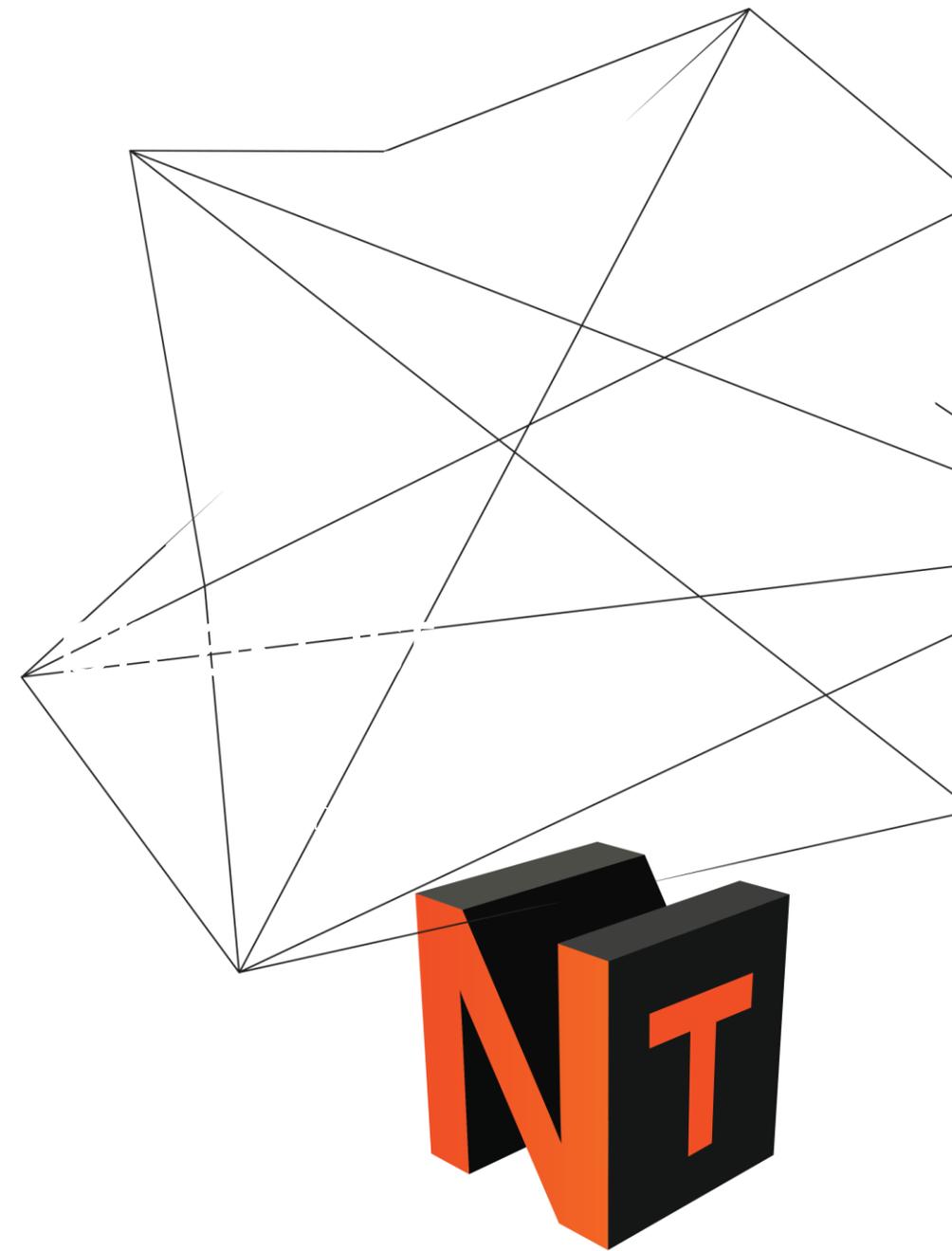
• Before



• After



The Teams Aligned Autonomy

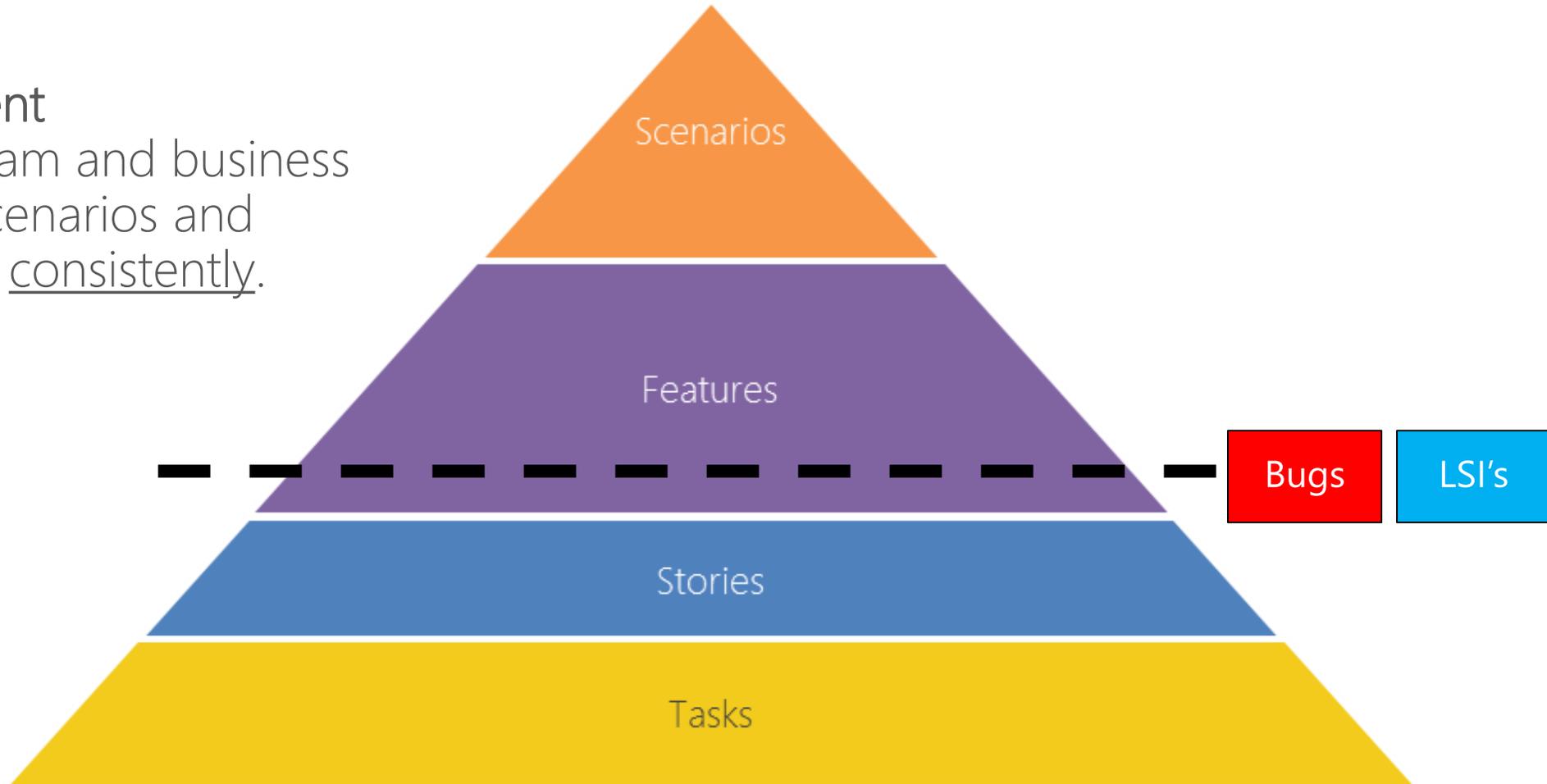


#ntk18

Taxonomy & Staying Aligned

Alignment

Every team and business tracks scenarios and features consistently.



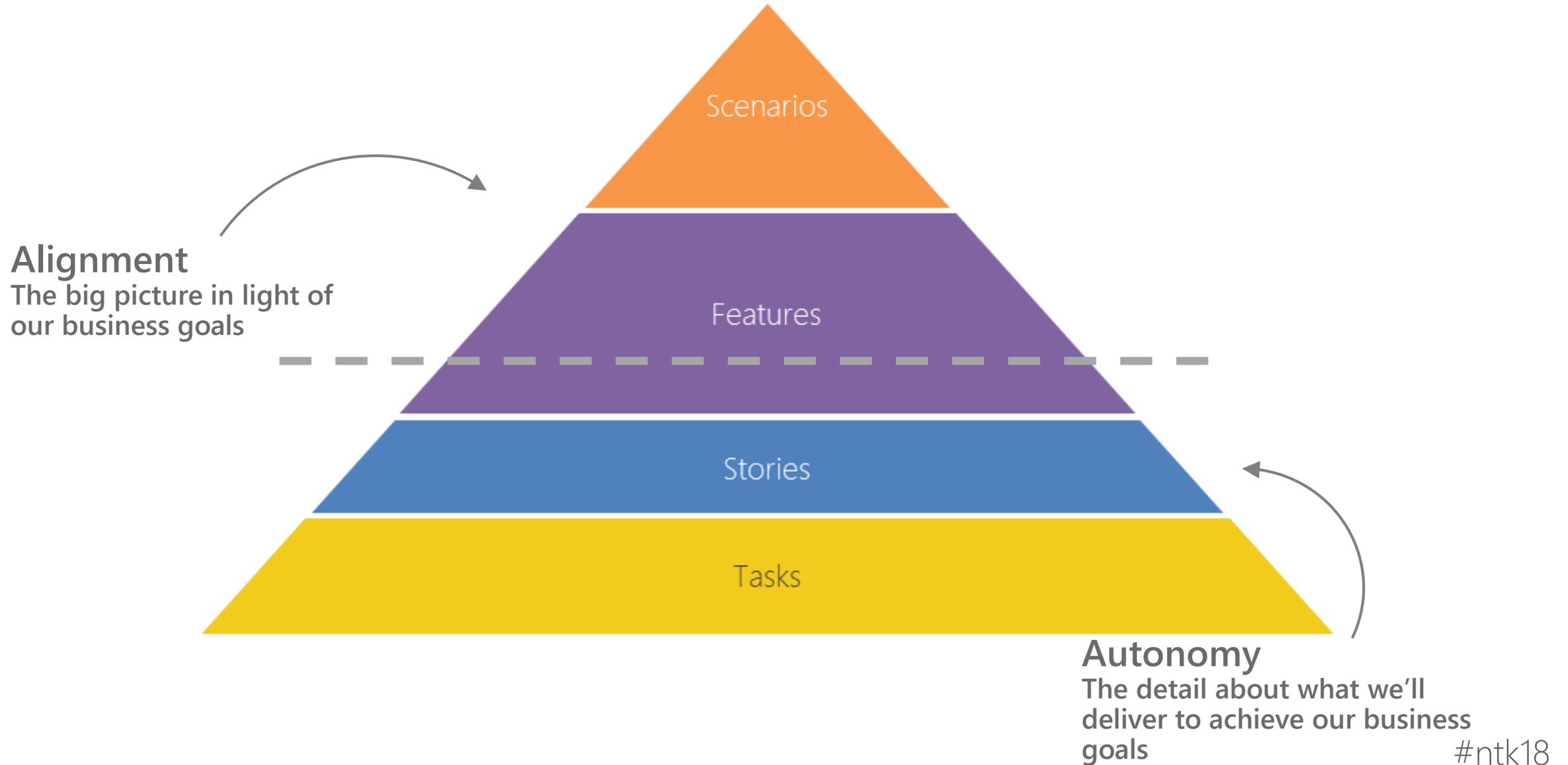
Autonomy

Every team chooses how to manage stories and/or tasks

#ntk18



Aligned Autonomy



Alignment

The big picture in light of our business goals

Scenarios

Features

Stories

Tasks

Autonomy

The detail about what we'll deliver to achieve our business goals

#ntk18



Agile

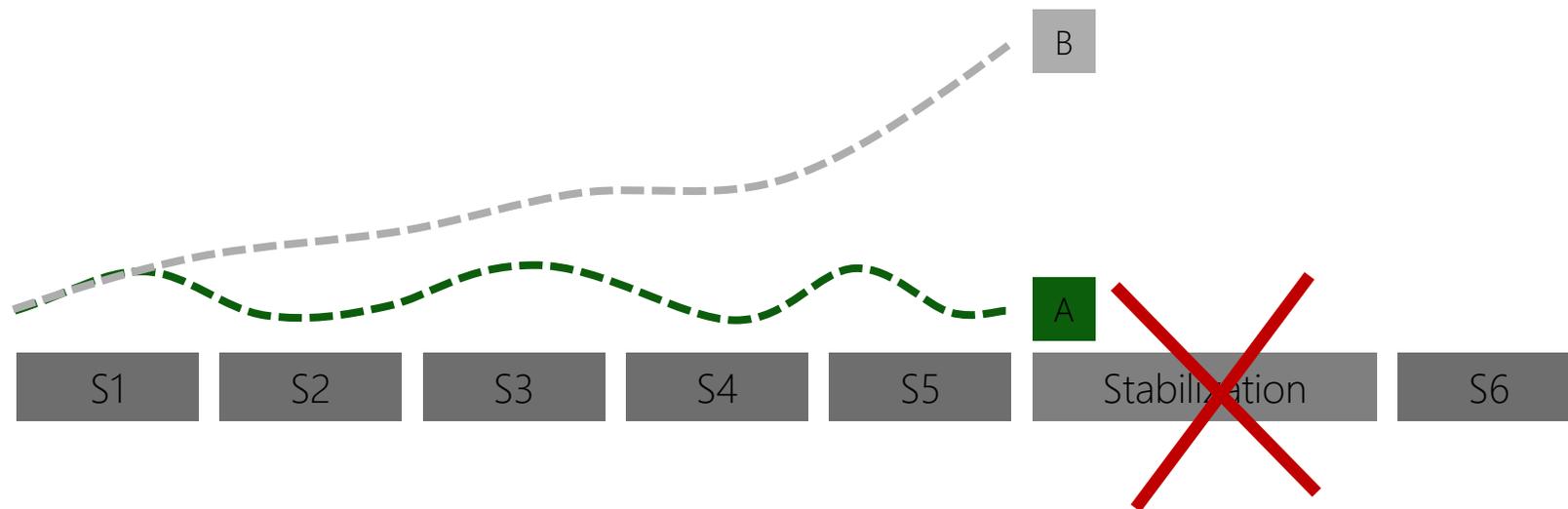
The schedule

No stabilization

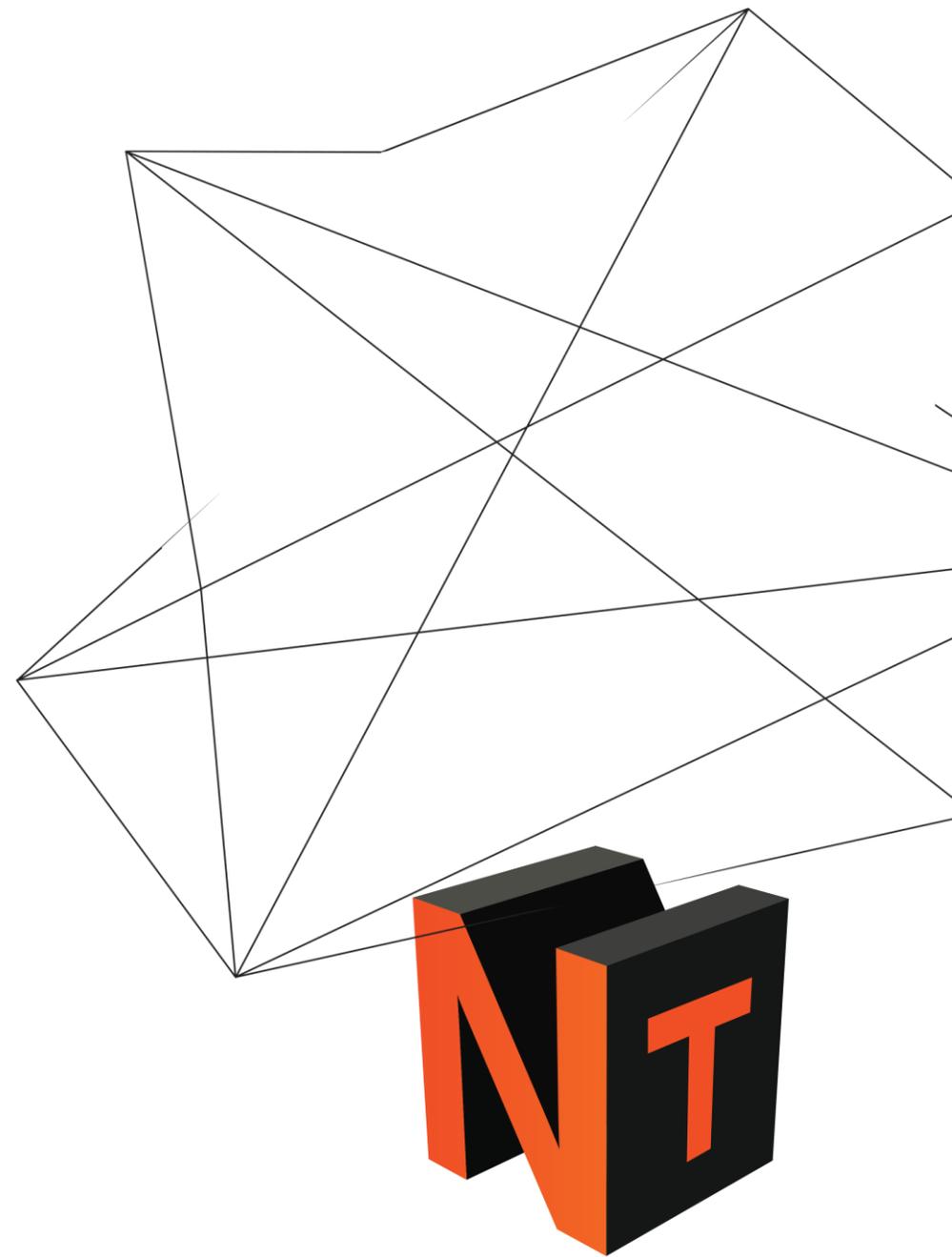
Alignment and Autonomy

You need both

Continuous planning (and learning)



The Teams have changed in many ways...



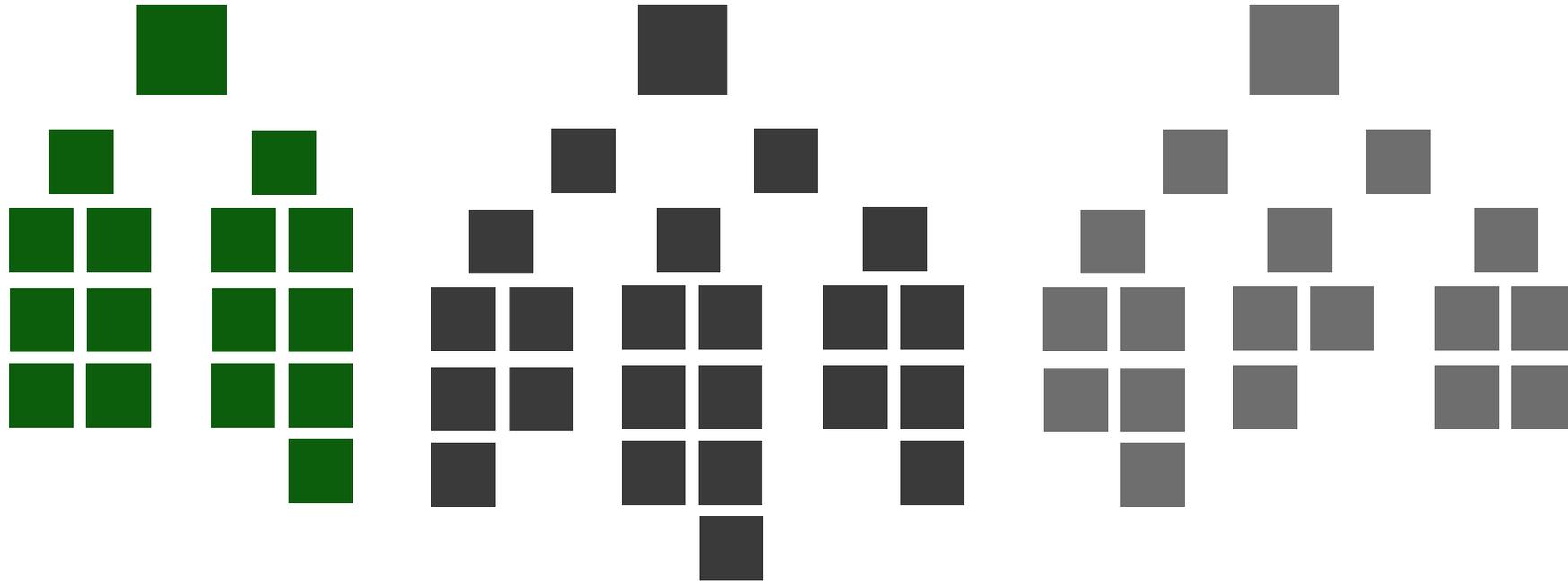
Roles

The OLD way

Program Management

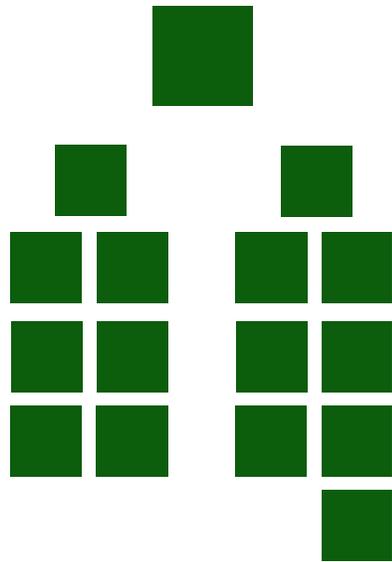
Dev

Test

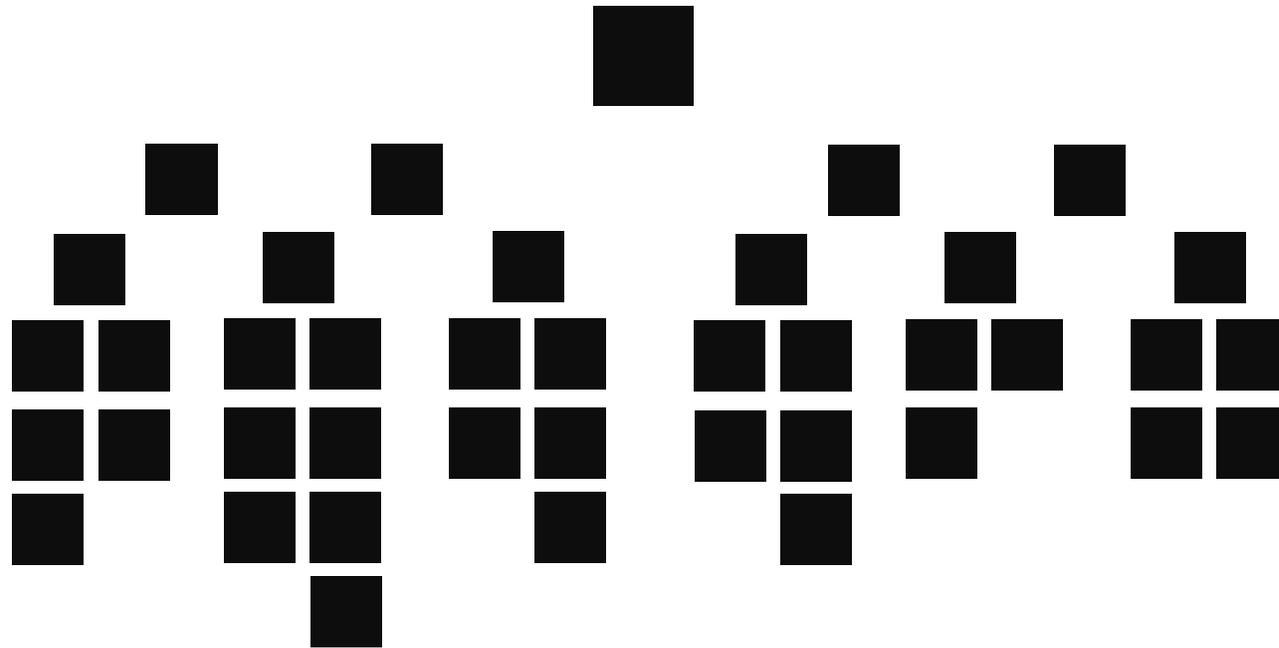


Roles

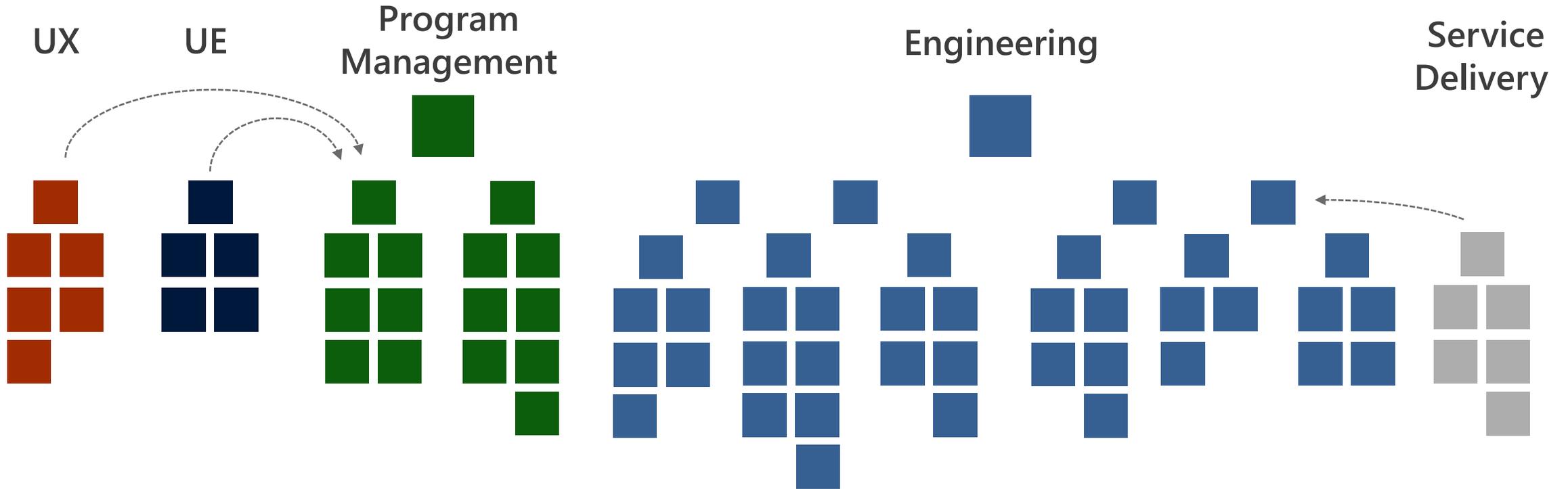
Program Management



Engineering



Yes, there are other roles...

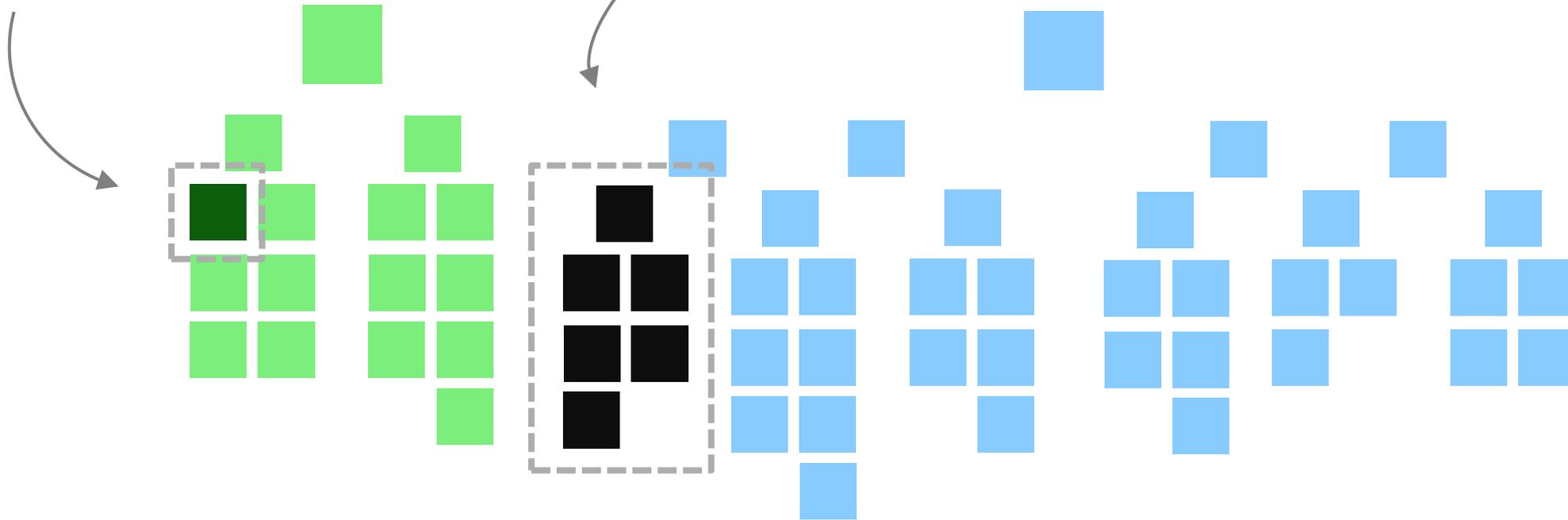


Service Delivery is integrated directly into development organization.

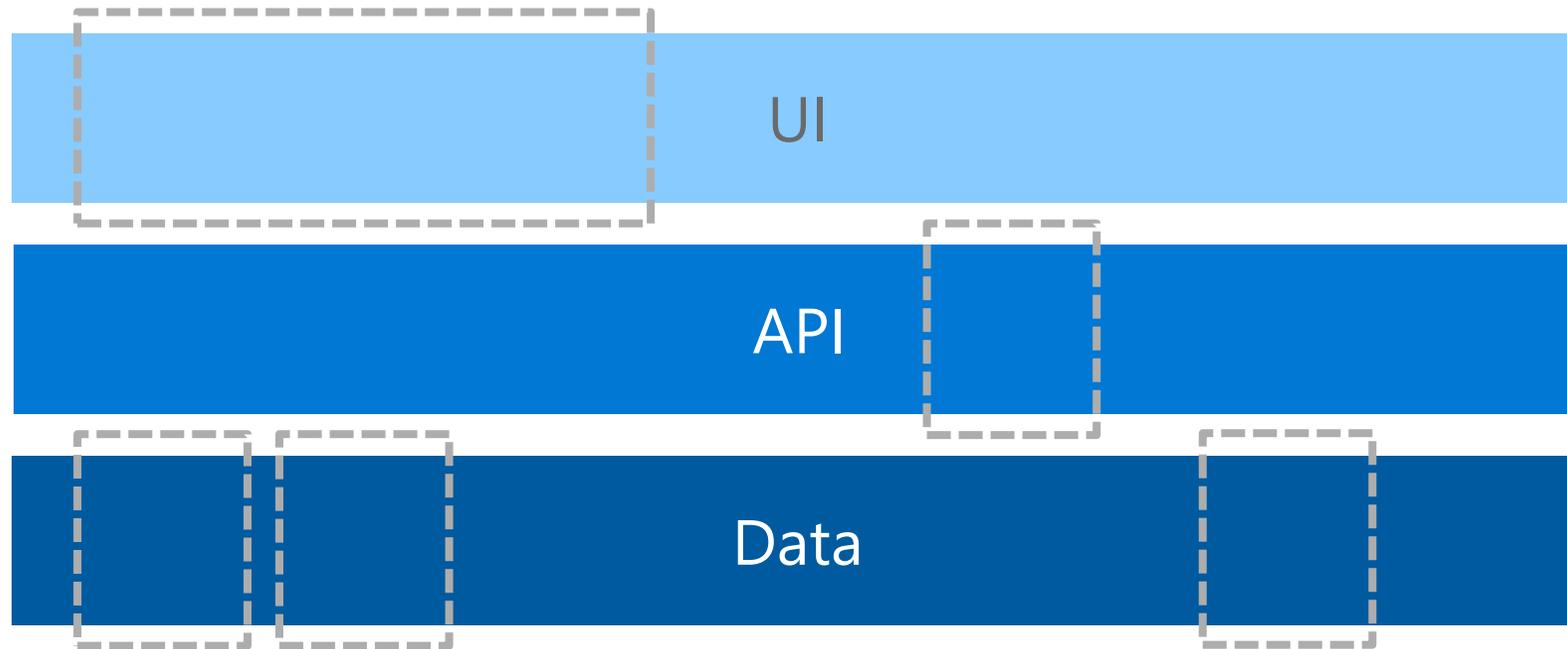
Teams

Program Management is responsible for:
WHAT they are building, and
WHY they are building it

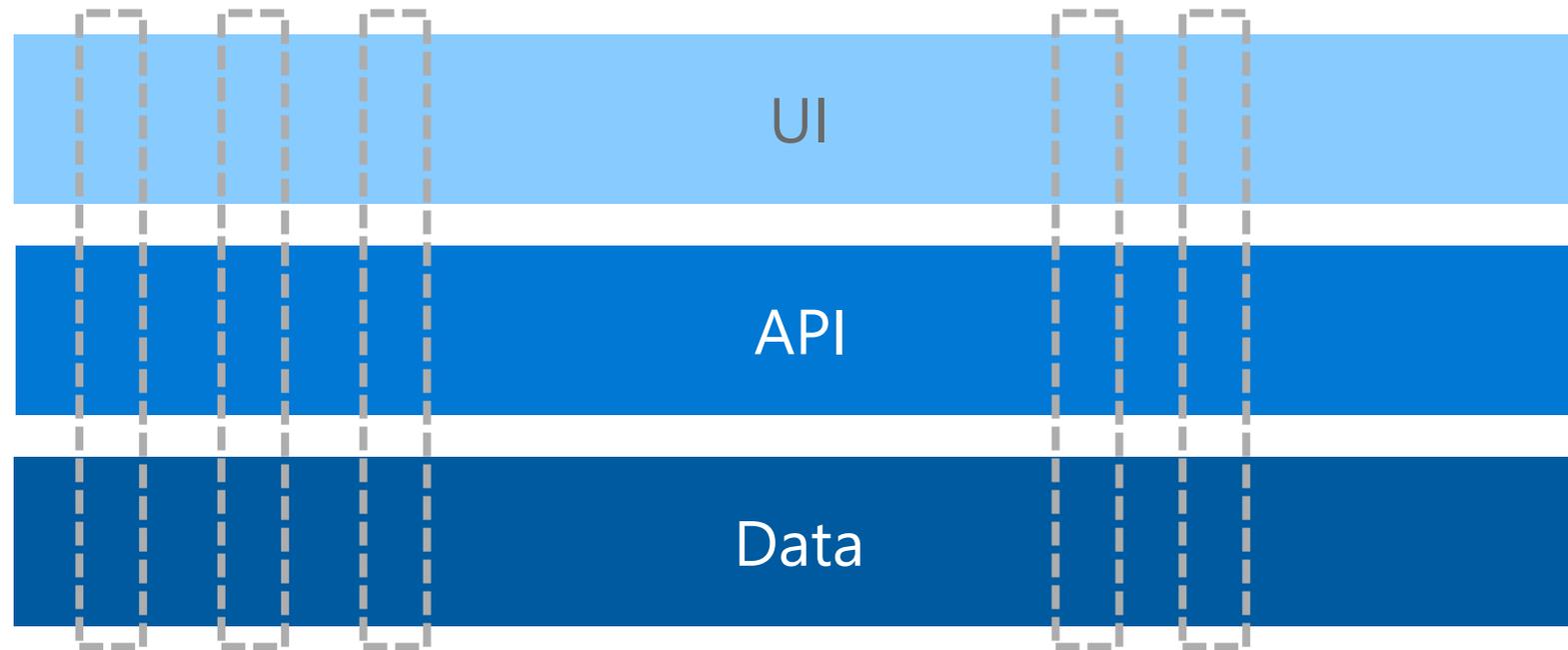
Engineering is responsible for
HOW they are building it, and that
they are building it with QUALITY



Before, Teams were Horizontal...



Nowdays, Teams Strive for Vertical



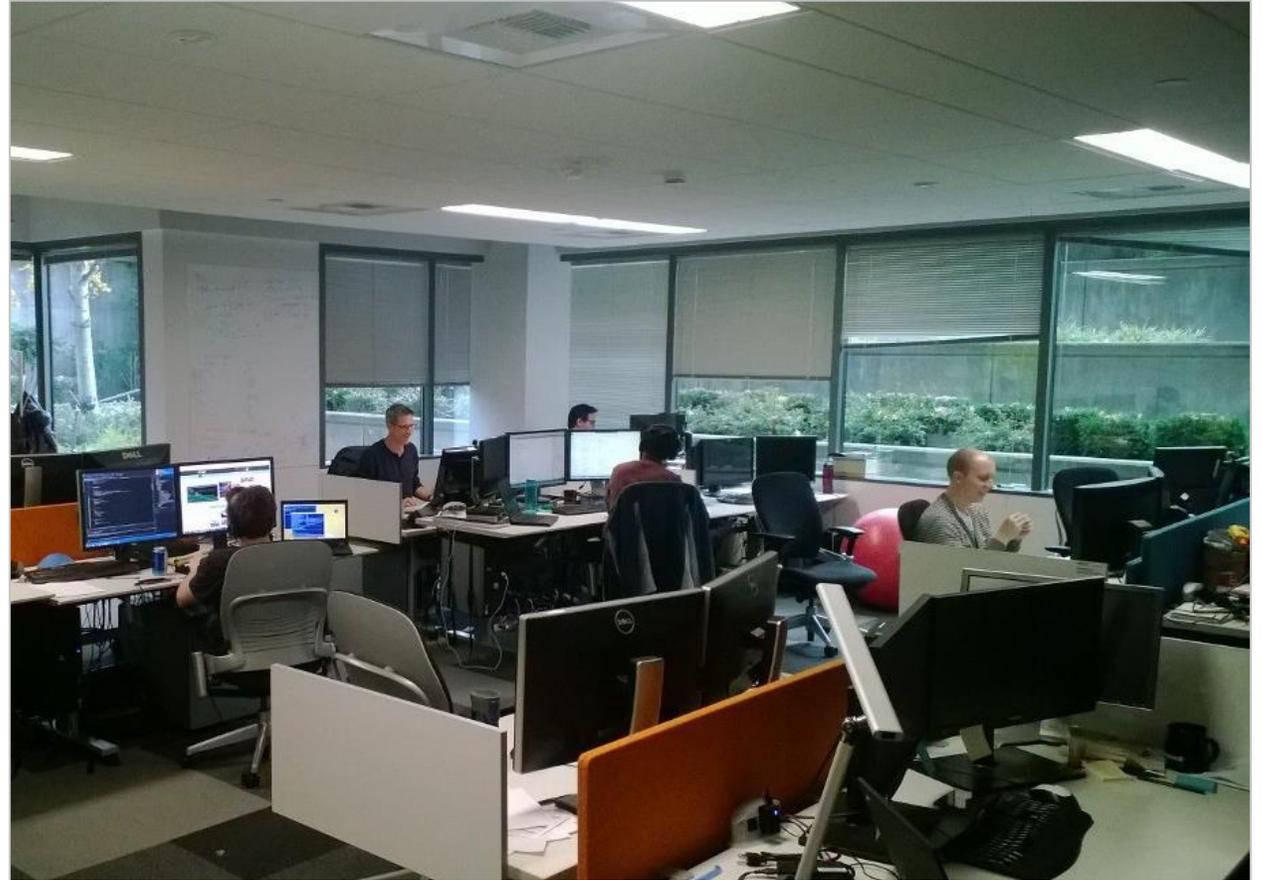
Teams

Cross discipline feature teams
10-12 people

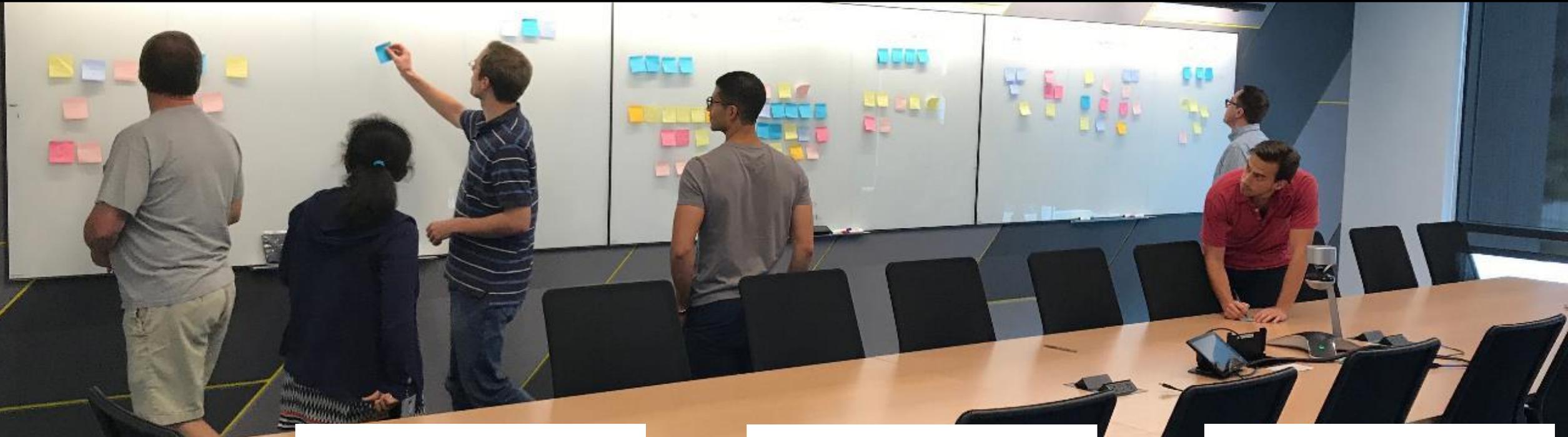
Self managing
Clear charter and goals

Own features in production
Own deployment of features

Physical team rooms
Intact for 12-18 months



Sticky Note Exercise - Self Forming Teams



Opportunity to change team without formal interviews or top down re-org

Employee choice, not manager driven

Unique approach within Microsoft

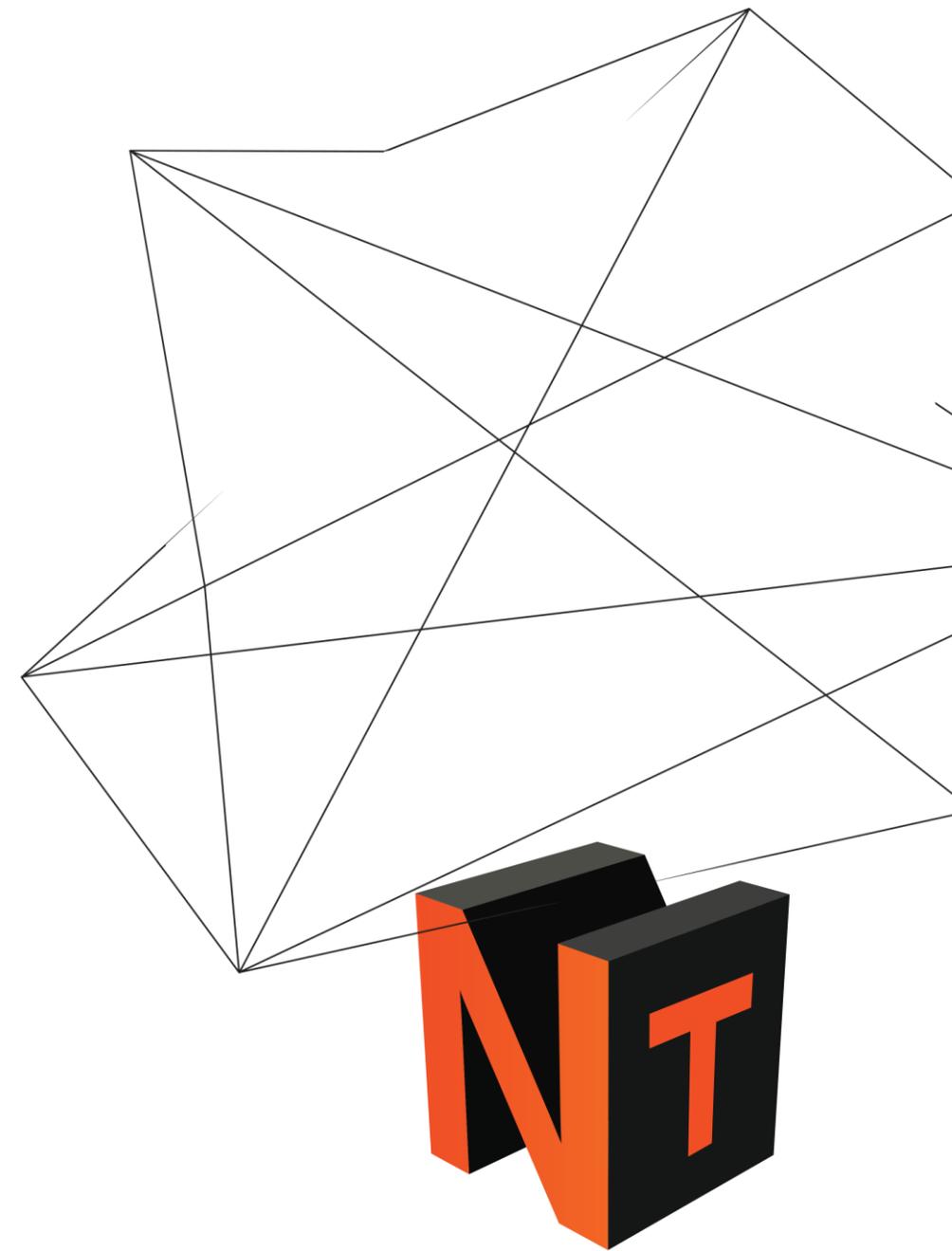
Typically <20% change, but 100% get to make a choice

Create opportunities for everyone to learn new things

Cross-pollinate talent and micro-culture



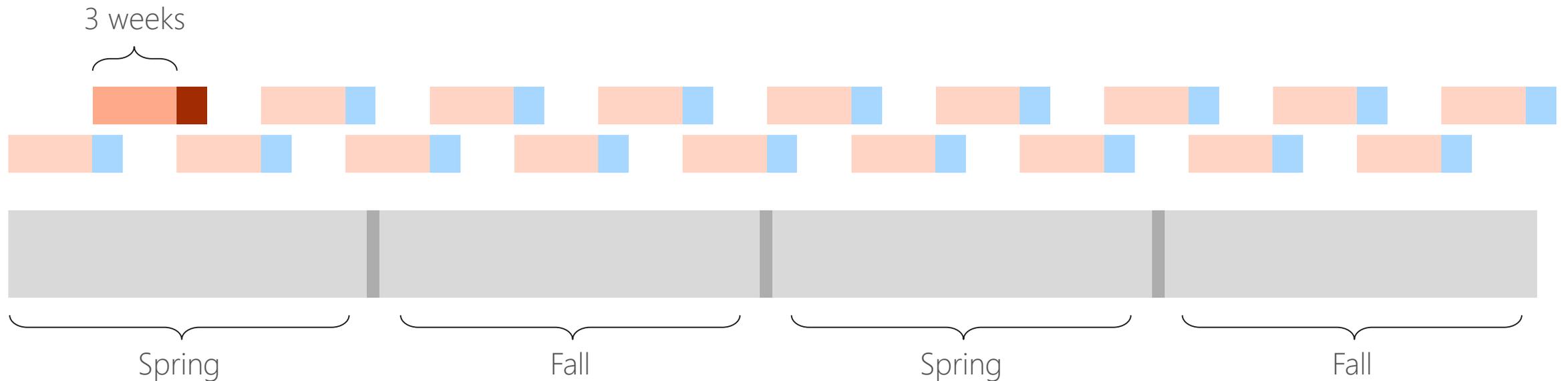
How do teams stay connected?



#ntk18

How do Teams stay in sync?

- 1. Sprint Mails – after each sprint**
2. Team Chats – after 3 sprints



Sprint Mails



Sprint 113



Sprint 114

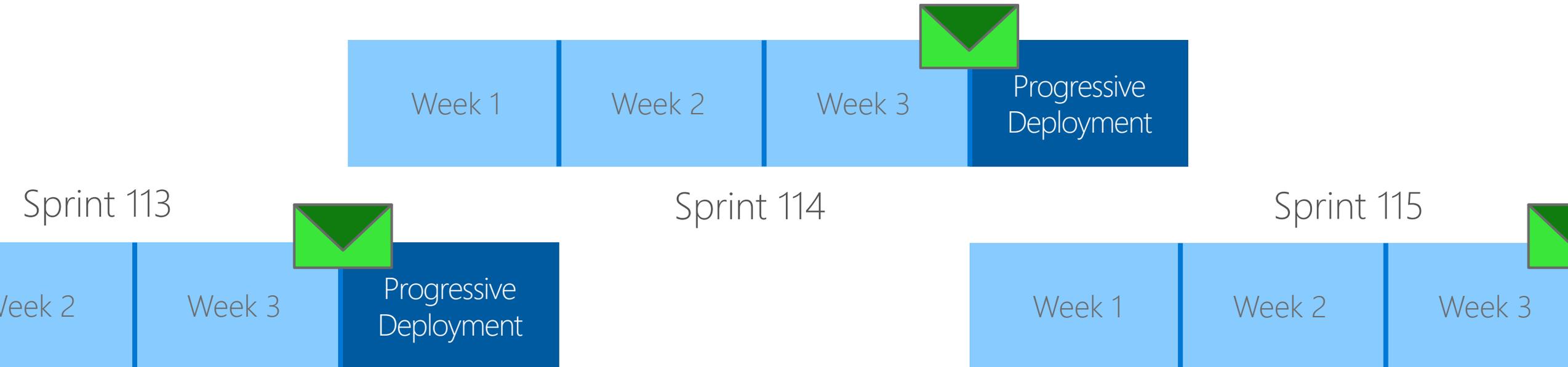


Sprint 115



Sprint Mails

At the end of a sprint, all teams send a “sprint mail”  ... communicating what they’ve accomplished in the sprint, and what they’re planning to accomplish in the next sprint.



Sprint Mails

Value delivered during the sprint

Video demonstrating the value

What the team is planning to accomplish in the next sprint

Blueprint S113 Summary/S114 Plan - Messag...

File Message Tell me what you want to do

Sondra Batbold VSTS Sprint Communication; VSTS Leadership; Blueprint

Retention Policy Never Delete (Never) Expires Never

You replied to this message on 2/14/2017 9:12 PM.

Sprint 113 Summary

Deliverables

- Updates to Plan settings no longer require full page refresh
- Fit and finish improvements to Delivery Plans:
 - Improved space and layout for team and backlog level
 - Added backlog links for quick access
 - Improved scroll behavior
- Field criteria allows users to tailor the plan view based on specific criteria (ex: only view cards with tag: Blocked) – will dogfood to Stage 0 with S113
- Addressed the bulk of the Accessibility bugs for the Kanban board as well as the Kanban settings experience
- Closed [45 bugs](#)

[Click to view video \(audio, 2 mins\)](#)

Blueprint - Sprint 113

By the numbers:

Agile - Blueprint

Team	Sprint	Progress
Blueprint Features	Sprint 113 (1/23 - 2/10)	2 Feature
Blueprint Stories	Sprint 113 (1/23 - 2/10)	52 Bug, 4 DTS Task, 20 User Story

Accessibility Burndown

Sprint 114 Plan

Deliverables

- Delivery plan:** Field Criteria allows users to tailor their plans by applying query clauses to the plan data set. [\[Spec\]](#)
- Accessibility:** wrap up Kanban settings and start Accessibility work for Delivery Plans
- Office Integration:** Diagnostics troubleshooting for Office integration bugs on MSDN
- Dev14 Update 4:** port over 3 DTS fixes and 3 Bug fixes
 - DTS Task 887828: DTS: TFS2015 @Today variable not working in French board styles
 - DTS Task 885626: DTS: TFS 2015.3 | Card styling rules doesn't apply on using @Heute (for @Today) in Deutsch (Deutschland) Locale.
 - DTS Task 878292: DTS: Work Item Title Not Showing in Chrome version 55 and Multiple Tasks Added
 - Bug 619011: Board styles issue in German
 - Bug 599181: Macro Ma alone is being evaluated and saved in localized format
 - Bug 797545: Title is not in edit mode when creating a new item from the board

New delivery plan

A delivery plan shows you when work will be delivered across your teams. The plan overlays each teams' sprint onto a familiar calendar view. You can view multiple backlogs and multiple teams across your whole account. [Learn more!](#)

Name: H2 Planning

Description: High priority items for the second half of 2017

Project: GoldPeak Team: GoldPeak Team Backlog: Stories

Field: Priority Operator: = Value: 1

Tags: CONTAINS Value: 2017 H2

Create

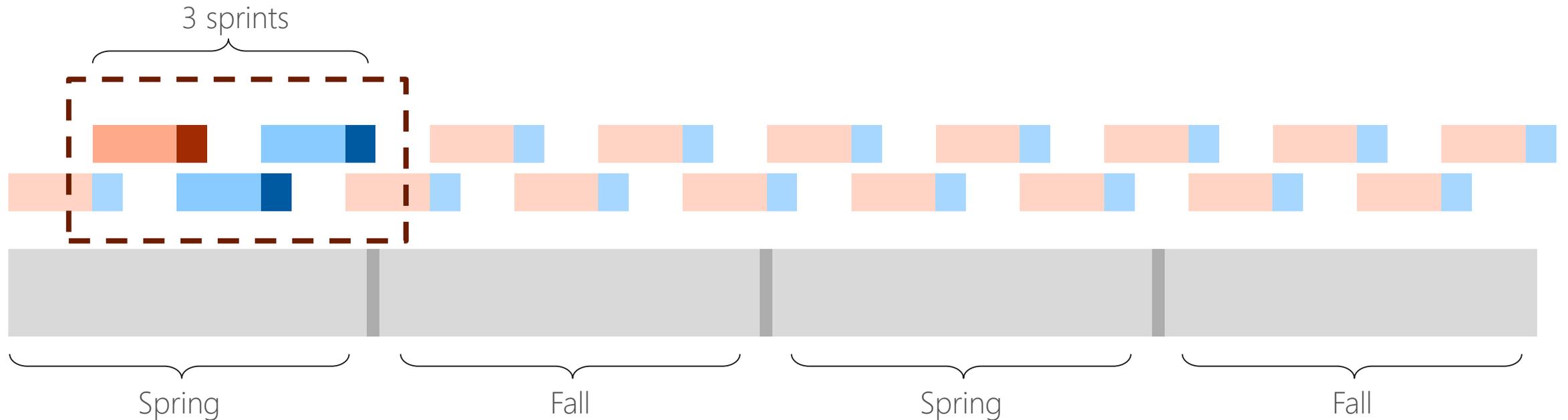
Customized plan showing priority and tag clauses

View our [Blueprint team Delivery Plan](#) for more details

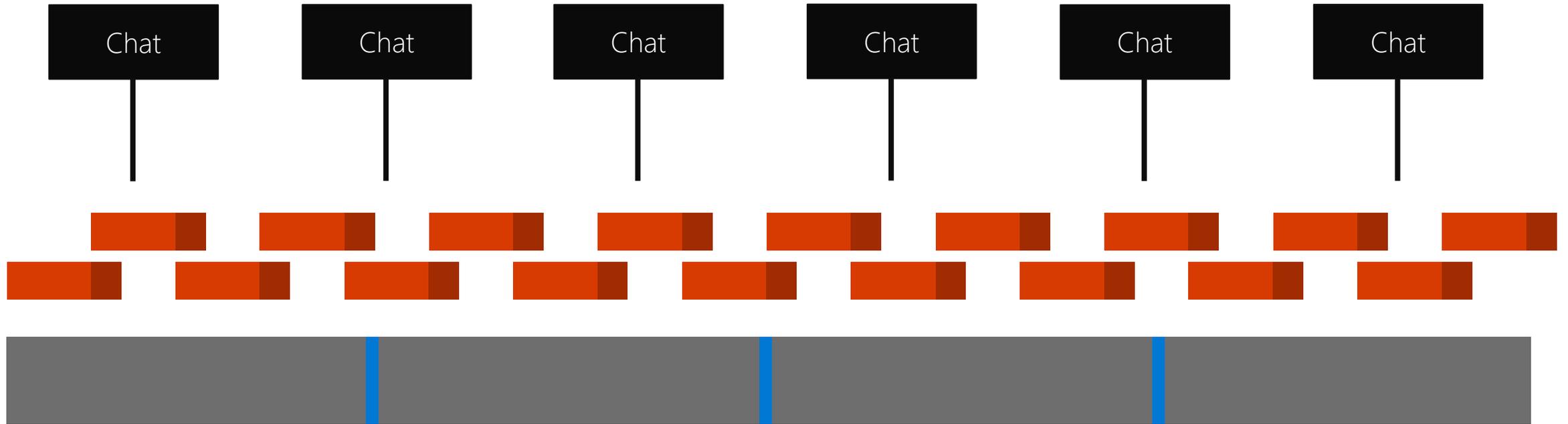


How do Teams stay in sync?

1. Sprint Mails – after each sprint
- 2. Team Chats – after 3 sprints**



Staying connected – Team “Chats”



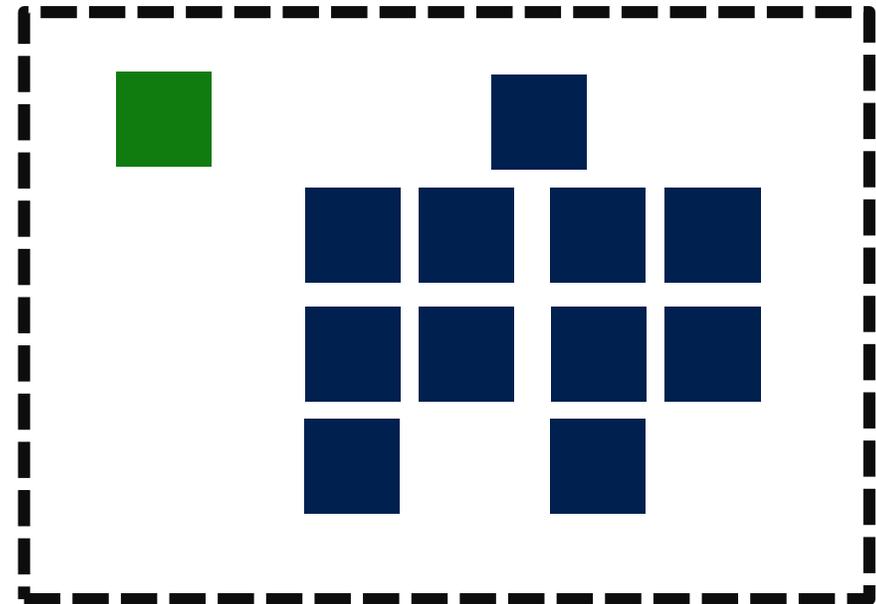
Every 3 sprints the leadership team sits down with each team for a “chat”



Team “Chats”

1. What’s next on your backlog?
2. How are you doing with regards to debt?
3. Any issues?

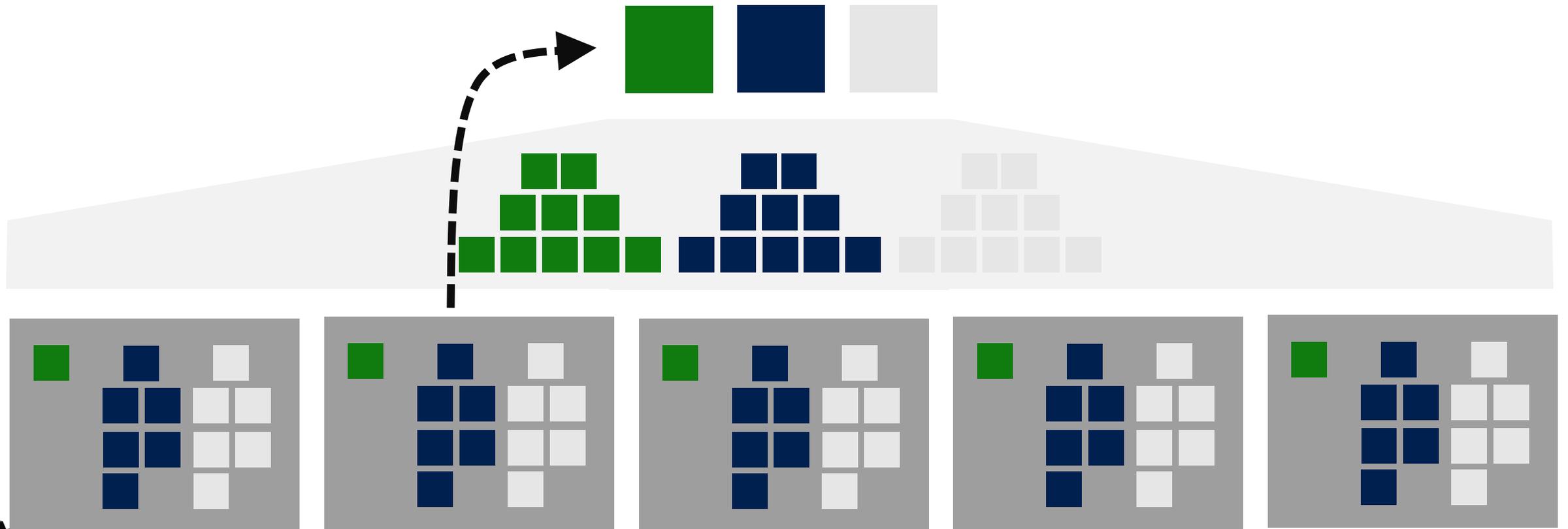
Version Control



Team "Chats"

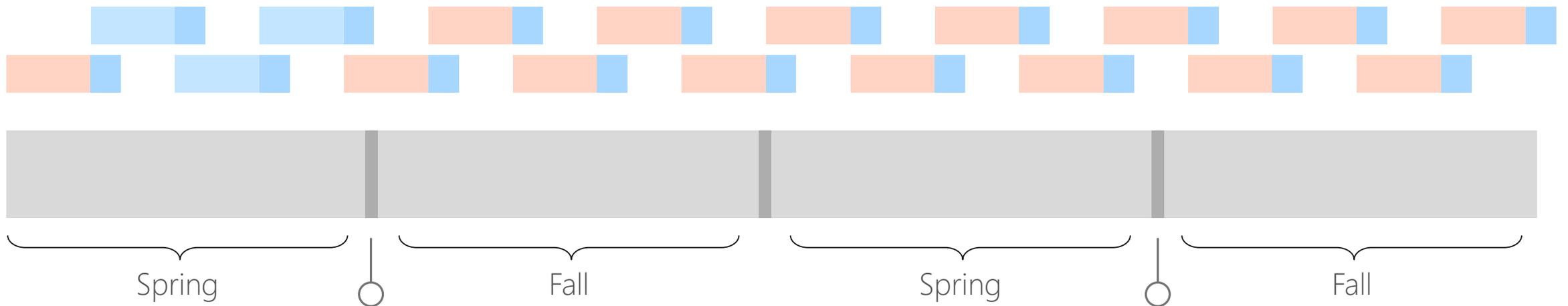


Team "Chats" Direct. No "lost in translation".

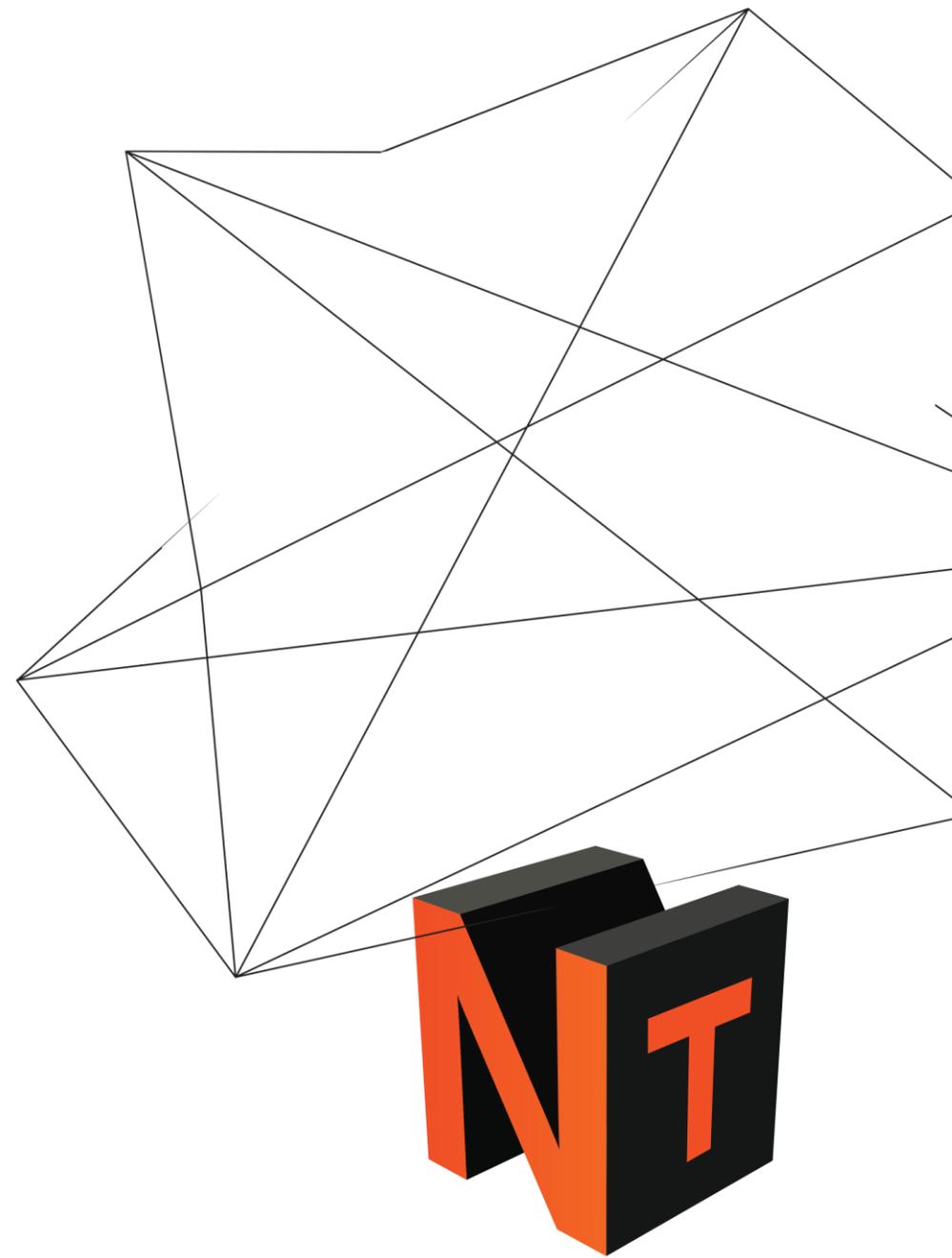


How do Teams stay in sync?

1. Sprint Mails – after each sprint
2. Team Chats – after 3 sprints

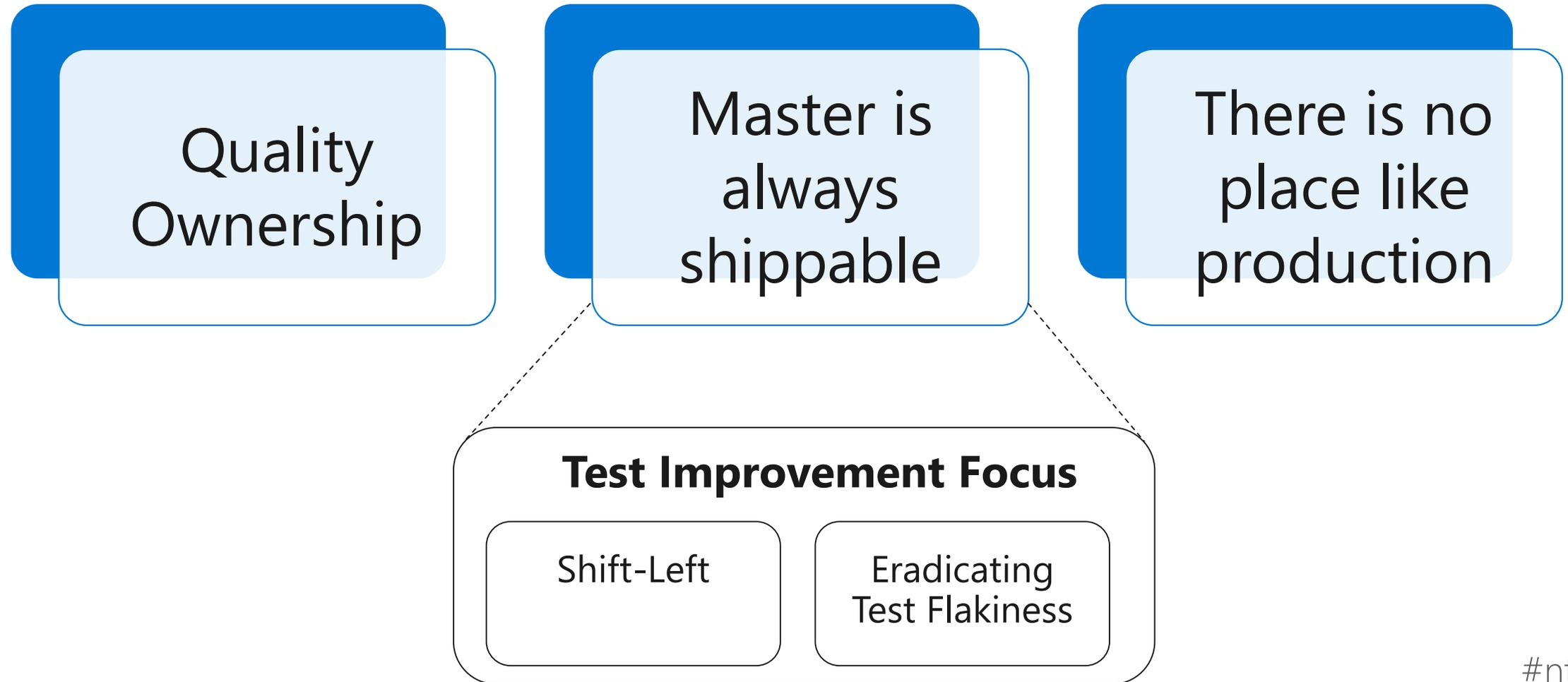


Quality in the Cloud Cadence



#ntk18

Quality in the Cloud cadence – What changed?



We combined the Dev and Test roles

New Engineer role merged responsibilities from dev and test

Everyone has a new role, everyone learning new skills

Every engineer and team has E2E accountability

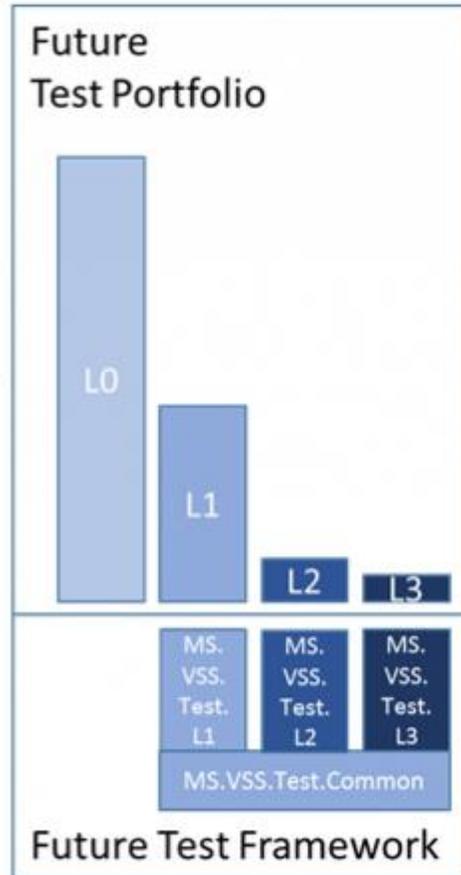
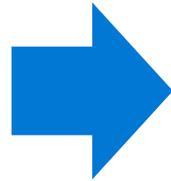
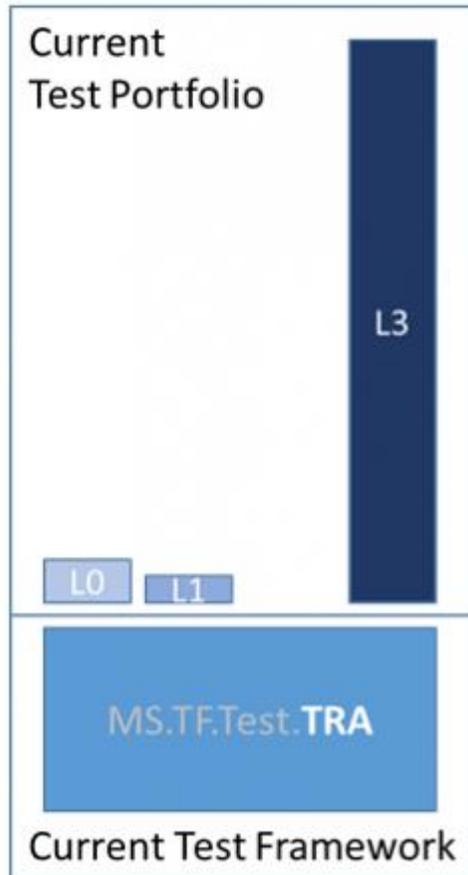
No more handoffs for testing, stronger emphasis on peer review

Big cultural shift across the company

Reducing number of people who have to agree and make a call
Increasing accountability and customer connection



Published VSTS Quality Vision : Feb '15

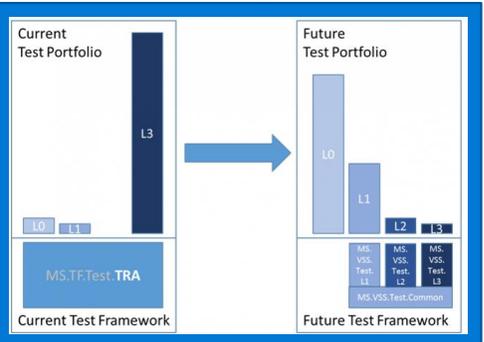
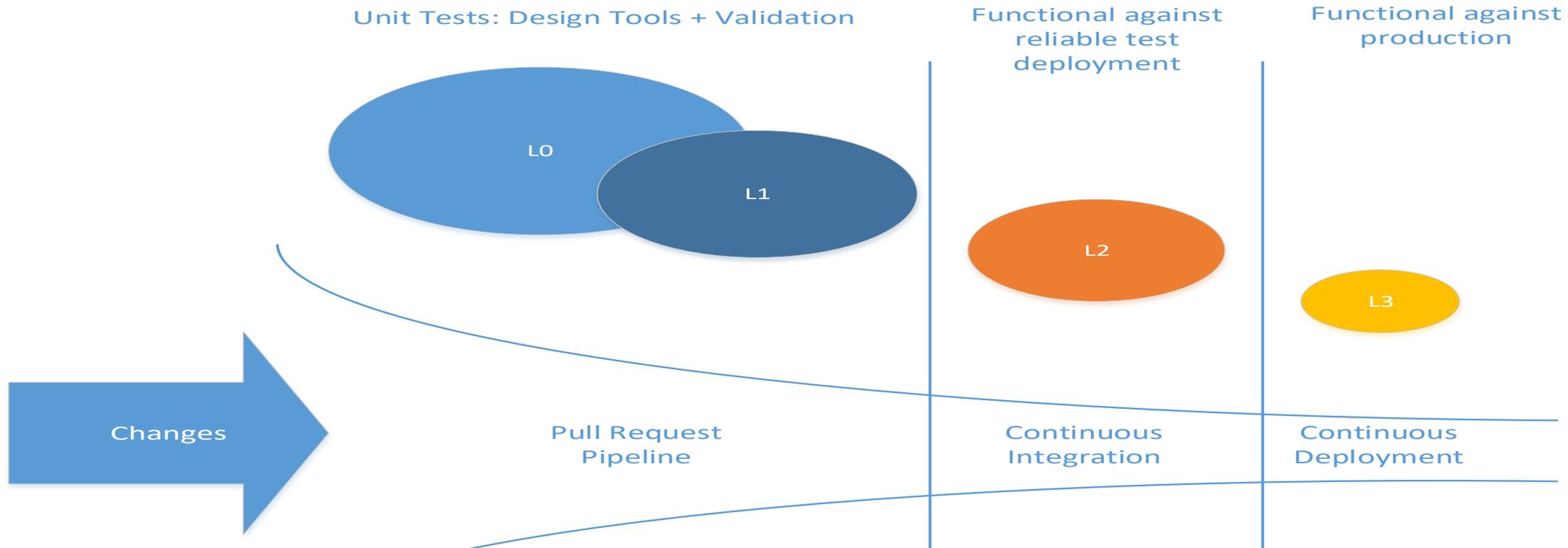


Principles

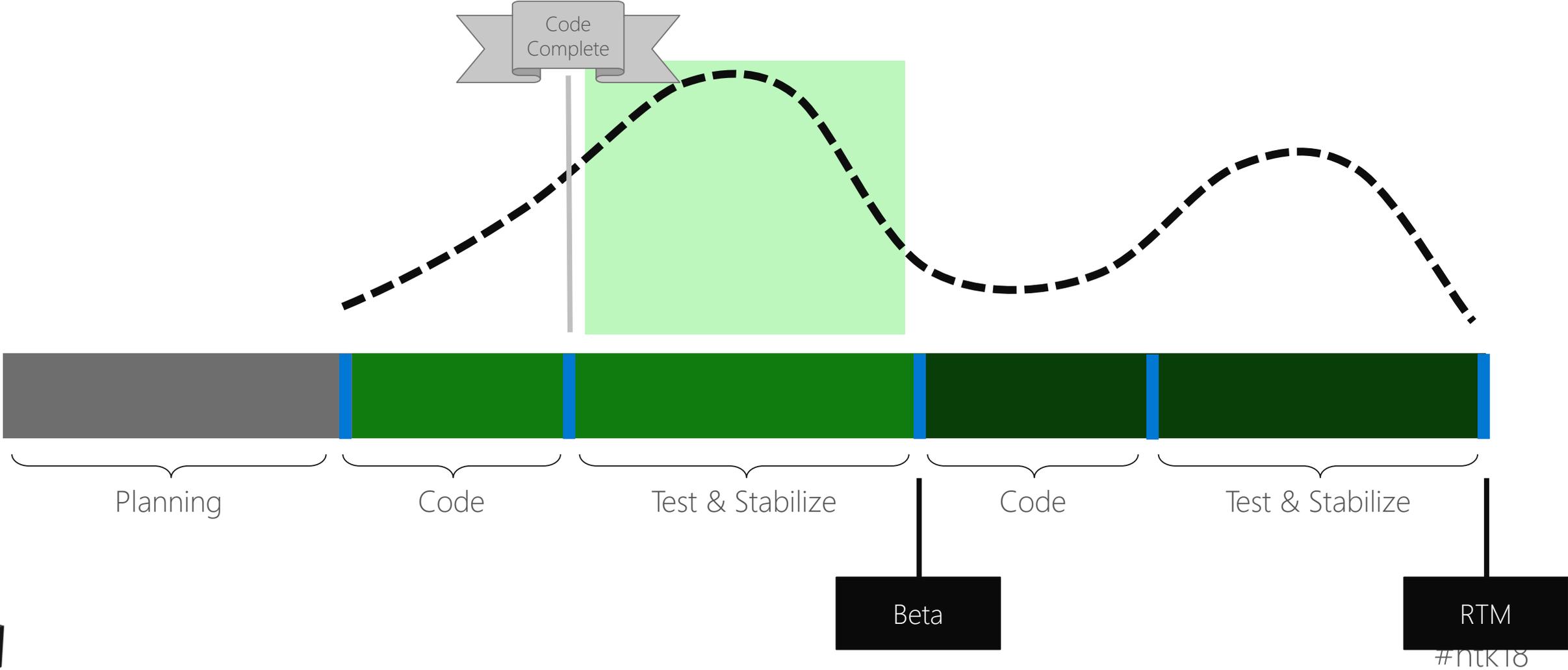
- Tests should be written at the lowest level possible
- Write once, run anywhere including production system
- Product is designed for testability
- Test code is product code, only reliable tests survive
- Testing infrastructure is a shared Service
- Test ownership follows product ownership



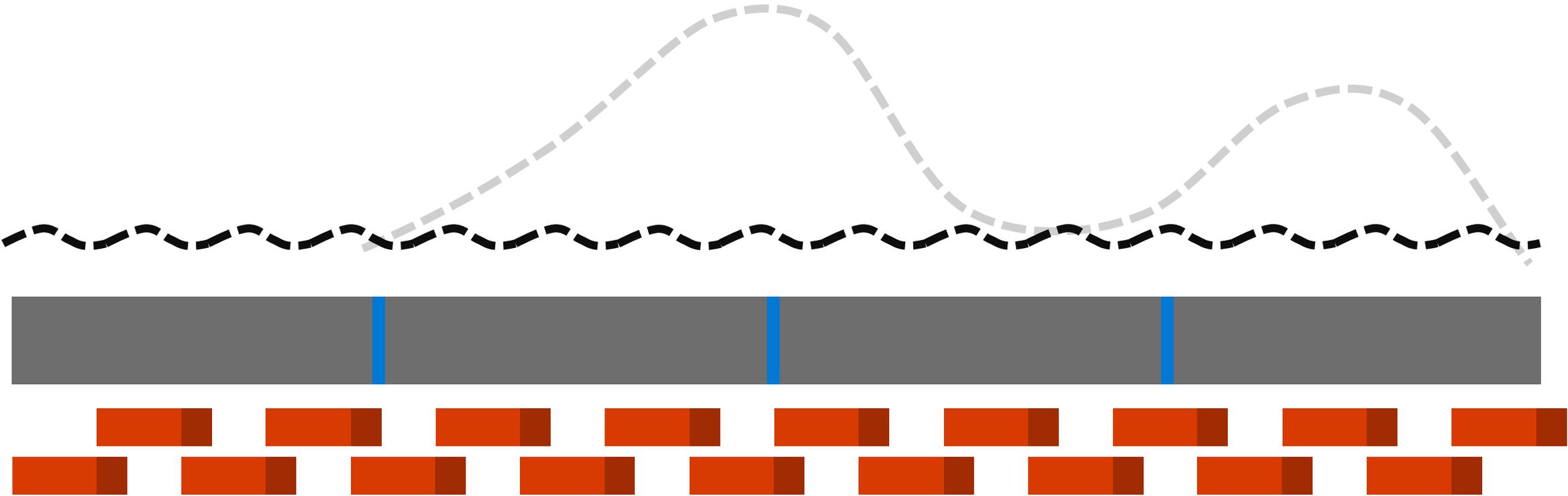
"Shift-Left" == Pushing Quality Upstream



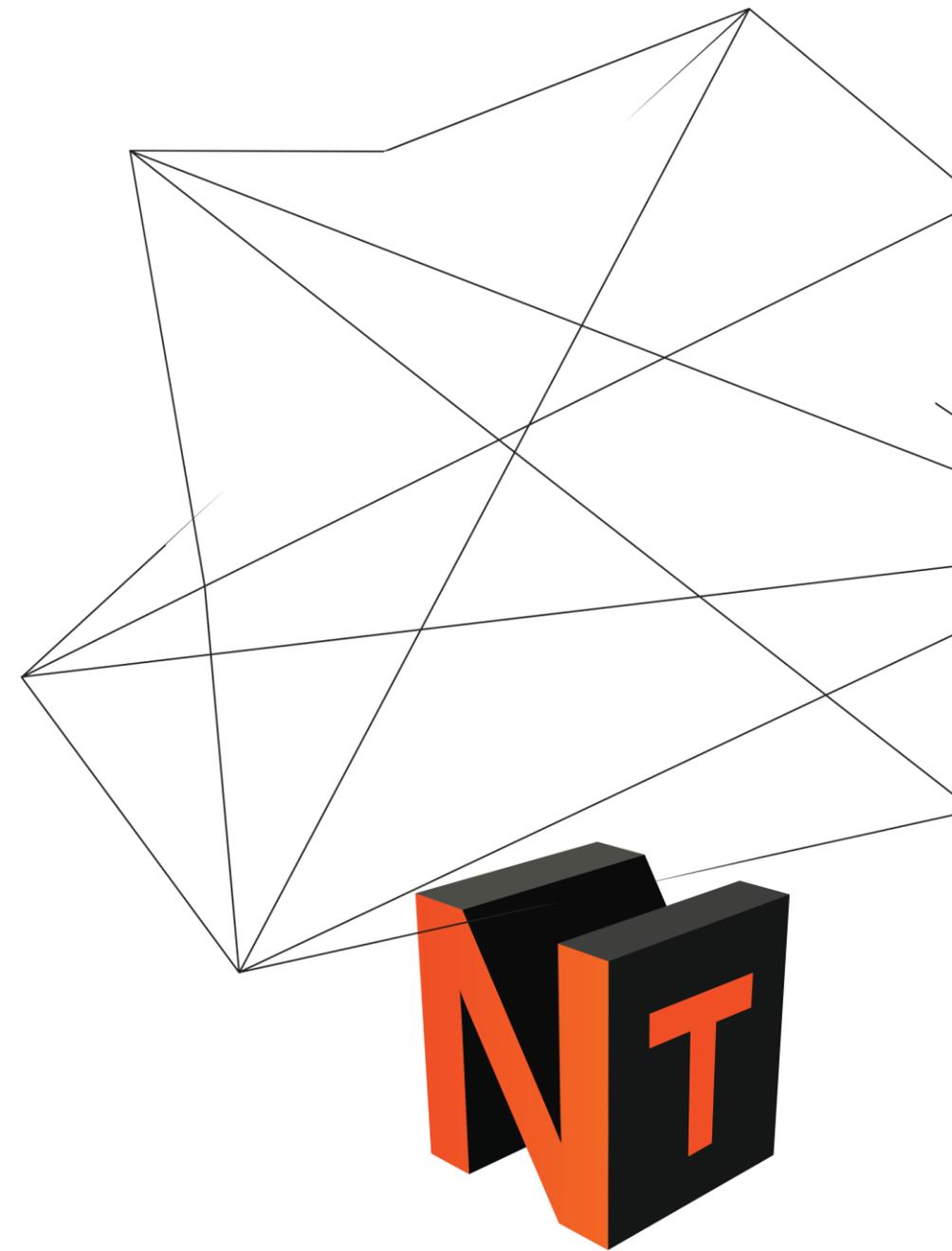
Quality- Before



Quality- After



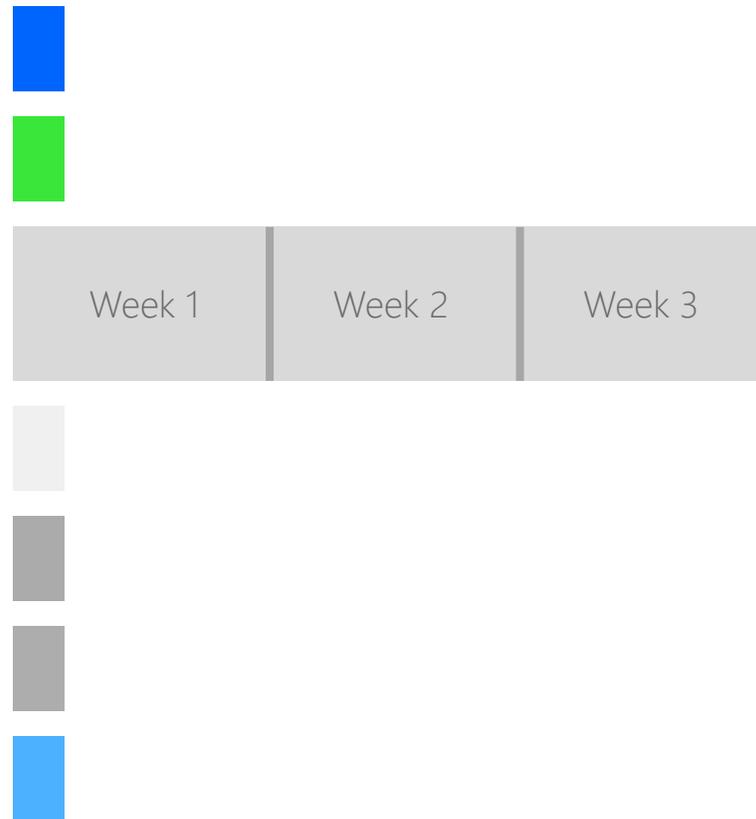
Day In The Life Of An Engineer



#ntk18

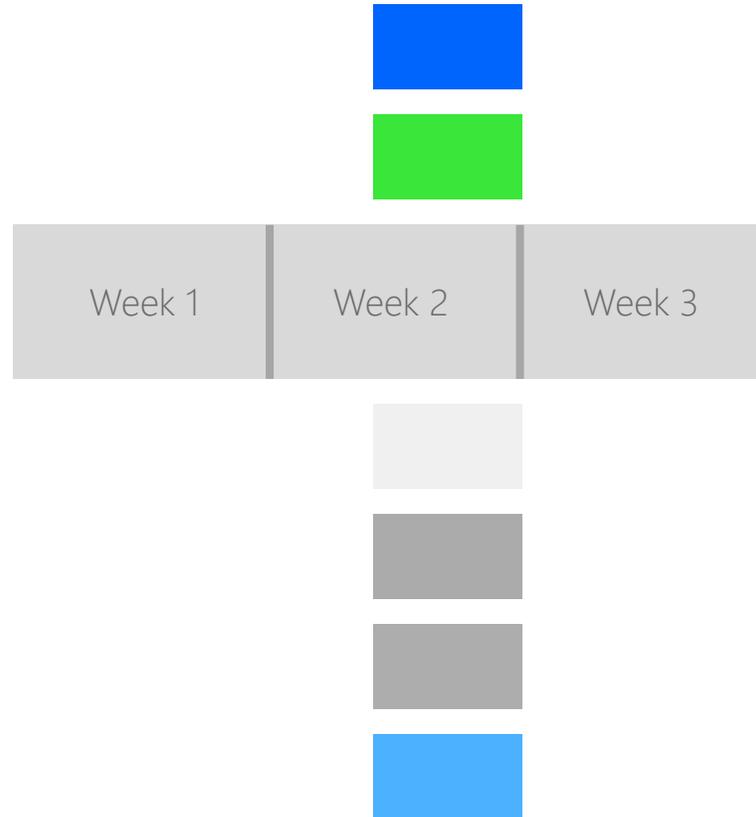
Everyone creates a branch...

The OLD way



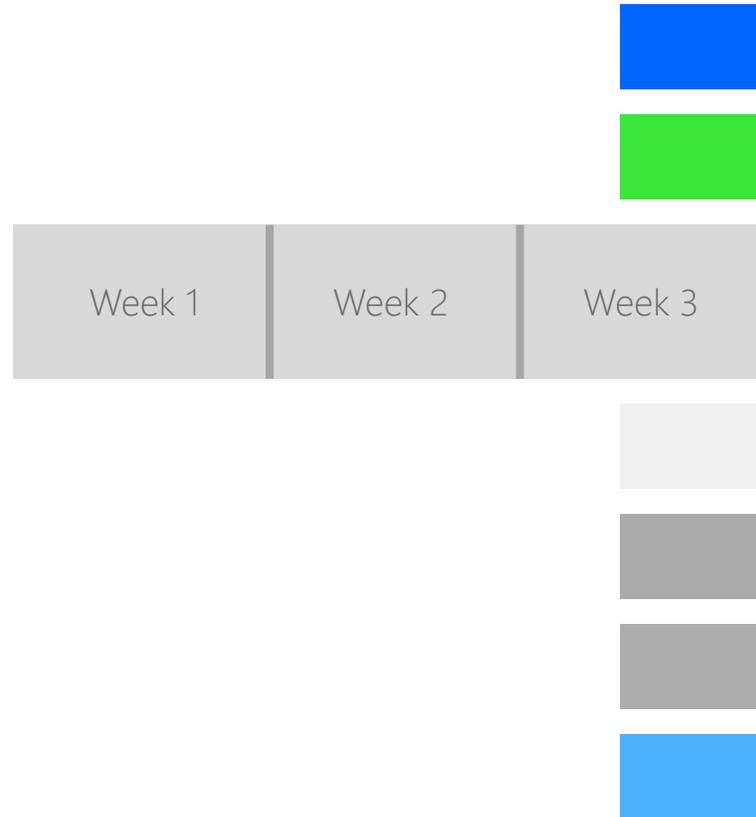
Writes a lot of code...

The OLD way



It needs to come together...

The OLD way



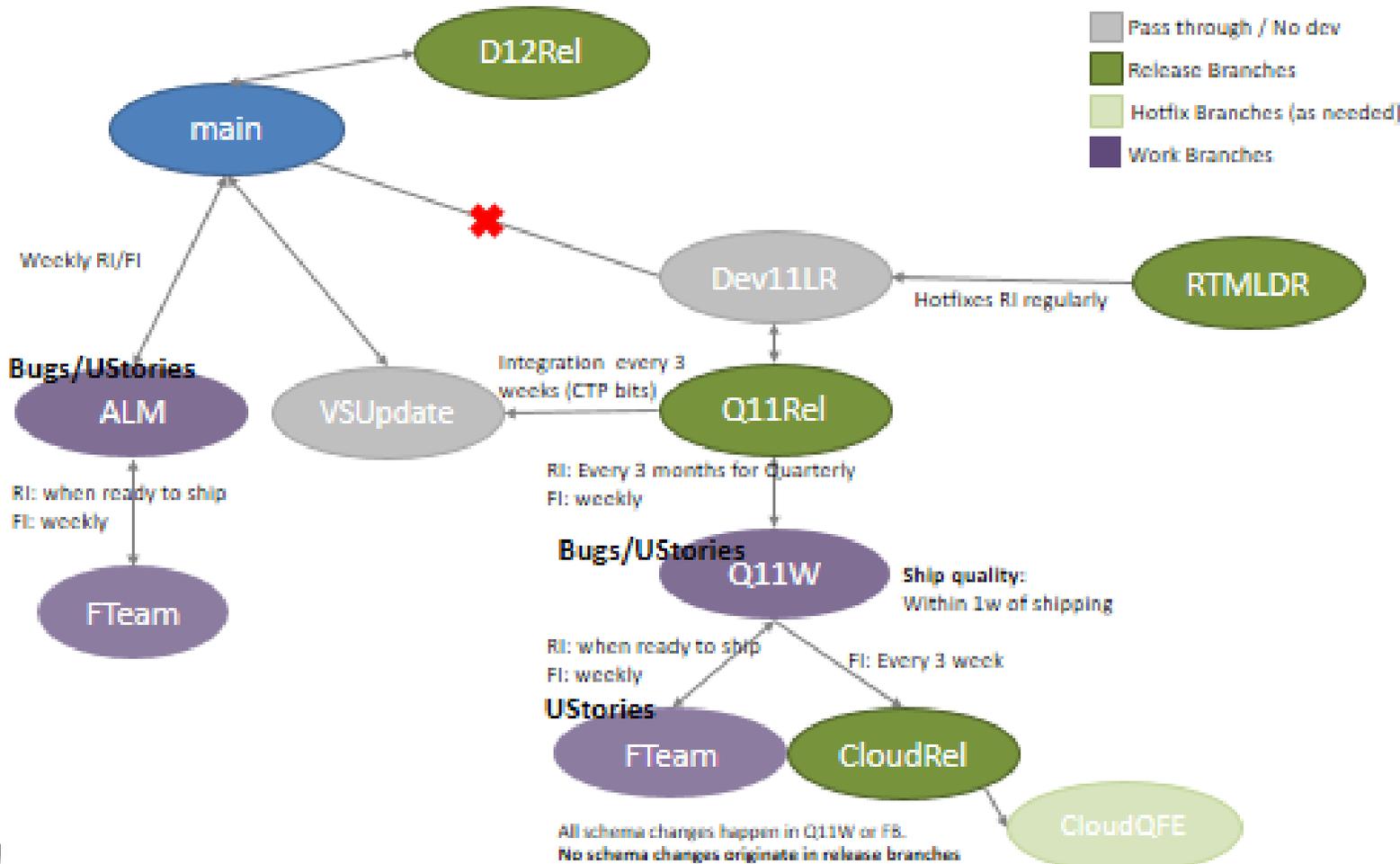
Merge Debt

The OLD way



Traditional branch structure

The OLD way



Deep branch hierarchy

Creates merge and integration debt

Significant costs to code flow

Complex logistics Engineers must understand



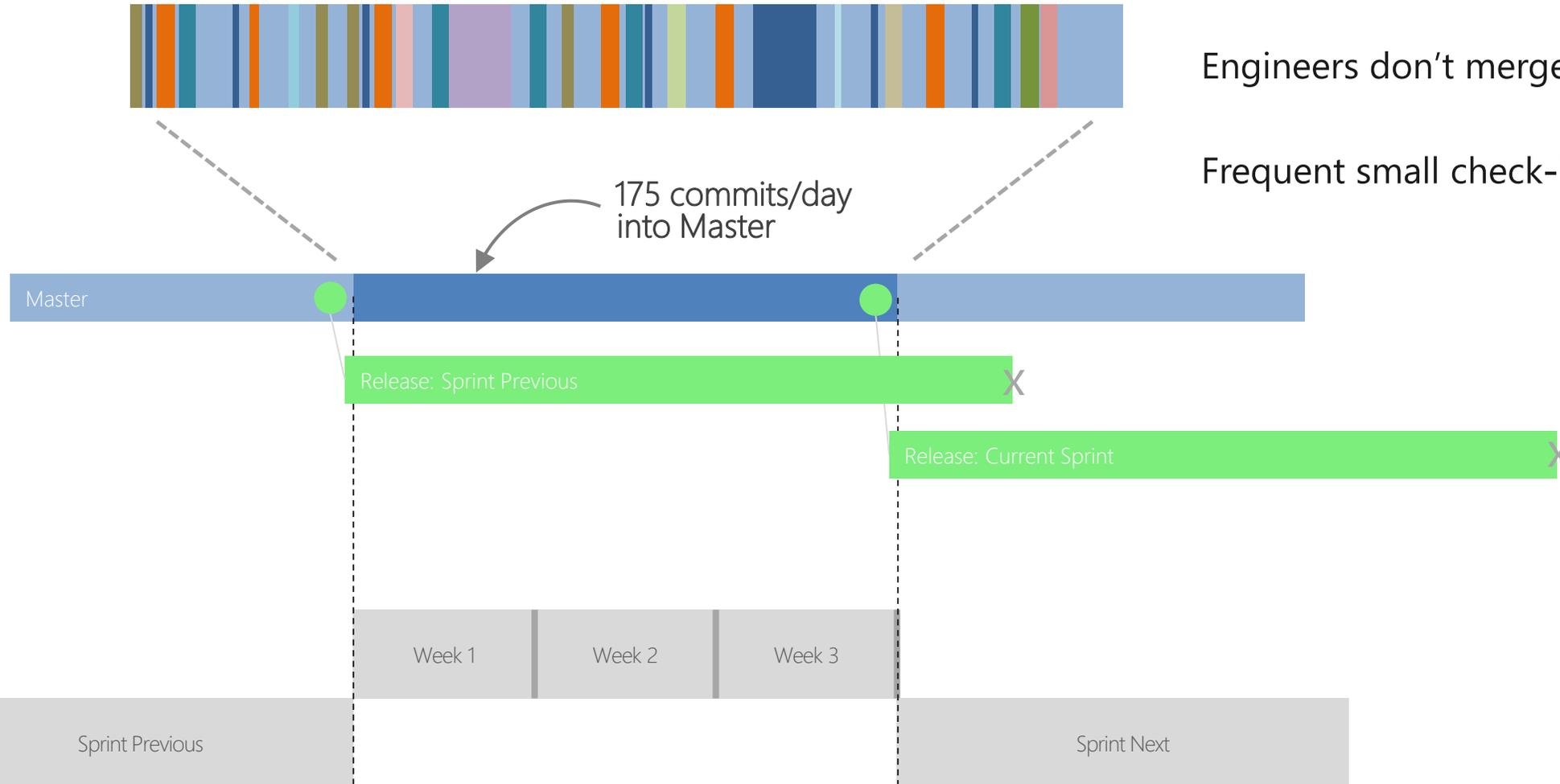
Work out of master

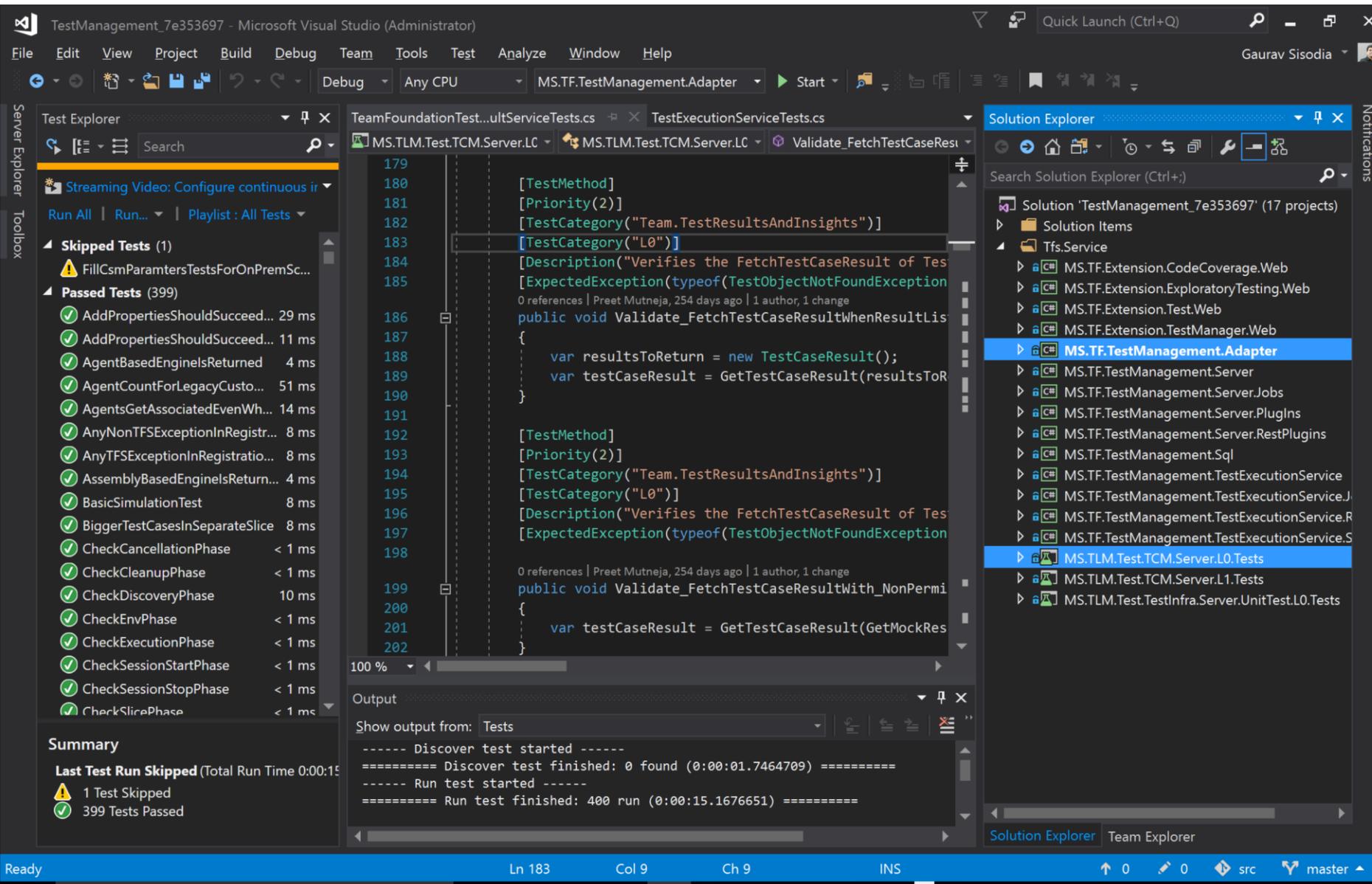
Source in a single git repo

All code flows to master

Engineers don't merge code they didn't write

Frequent small check-ins





Visual Studio, VS Code typically used for development

Test code collocated with product code

Author and run tests during development

Fast and reliable tests promote fear free refactoring

Build less, smile more! Try the new 'bp' command to do a partial build. bp saves time by building only what's needed to deploy specified products. Example:

```
bp mps.L2 sps.L2      <-- Builds the mps.L2 and sps.L2 products
vssdf mps.L2 sps.L2  <-- Deploys the mps.L2 and sps.L2 products
```

```
svajjala@SVAJJALA-SRV C:\VSO\src
$ vssdf sps.l2 tfs.l2 gallery.l2
```

Step	Time
-----	-----
Install-RedisService	00:00:10.8461159
Provision-ServiceAzureResources	00:00:07.3070555
New-ServiceConfigDb	00:06:33.0678598
Install-ServiceBinaries	00:02:02.9530881
Wait-DefaultPartitionDatabase	00:00:04.2342929
Register-Dns	00:00:17.1162215
Assert-InheritedDefinitionsRegistered	00:00:04.7520756

Step	Time
-----	-----
Reserve ports	00:00:01.2352776
Preparing SQL Server	00:00:19.4071232
sps.L2	00:03:52.2297572
Gallery.L2	00:06:01.4090399
tfs.L2	00:09:23.6893882
Total	00:20:17.1495662

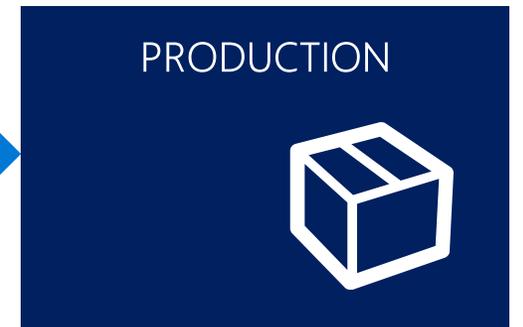
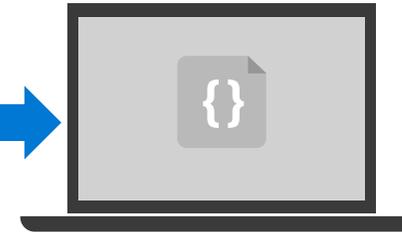
```
svajjala@SVAJJALA-SRV C:\VSO\src
$
```

PowerShell for service management

Used in production through Release Management workflows



Deployment/layout authoring in source tree



Gallery.Server.L2.Tests - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help

Quick Launch (Ctrl+Q) Sai Krishna Vajjala

Test Explorer

Run All | Run... | Playlist: All Tests

Passed Tests (2)

- ✓ PublisherCreateUpdate 13 sec
- ✓ TestTokens 3 sec

Not Run Tests (13)

- ! CreateAndRotateKeys
- ! ExtensionAssets
- ! ExtensionChangeScopes
- ! ExtensionCreateRestricted
- ! ExtensionCreateUpdate
- ! ExtensionPublicPrivate
- ! ExtensionQueryAdvanced
- ! ExtensionSecurity
- ! ExtensionStatisticsQuery
- ! ExtensionTrusted

PublisherCreateUpdate Copy All

Source: PublisherUnitTests.cs line 22

✓ Test Passed - PublisherCreateUpdate

Elapsed time: 0:00:13.0095863

Output

```
13
14 namespace Gallery.Server.L2.Tests
15 {
16     [TestClass]
17     0 references | Philip Kelley, 120 days ago | 1 author, 1 change | 1 work item
18     public class PublisherUnitTests : GalleryServerTestBase
19     {
20         [TestMethod, Owner("vishnupriyan"), Priority(1)]
21         [Description("Verify that we can create publishers with a variety of states.")]
22         ✓ | 0 references | Philip Kelley, 120 days ago | 1 author, 1 change | 1 work item
23         public void PublisherCreateUpdate()
24         {
25             ServerTestHost collectionTestHost = ServerTest.DefaultHost;
26
27             using (IVssRequestContext requestContext = collectionTestHost.CreateUserRequestContext())
28             {
29                 IPublisherService publisherService = requestContext.GetService<IPublisherService>();
30                 Publisher publisher;
31
32                 // Should Fail to create a new publisher with the same name.
33                 AssertHelper.Throws<ArgumentException>(() => publisherService.CreatePublisher(requestCo
34                 AssertHelper.Throws<ArgumentException>(() => publisherService.CreatePublisher(requestCo
35
36                 // Generate a random Publisher and make sure we can create it.
37                 String publisherName = Guid.NewGuid().ToString().Replace("-", String.Empty);
38
39                 // Create a random publisher with our generated named, owned by our generated user.
40                 publisher = publisherService.CreatePublisher(
41                     requestContext
```

Quality shift-left

Run and debug functional tests (L2/L3) in VS

VSOnline / VSOnline Te... | Dashboards Code Work Build and Release Test ... | Search this project

VSO Files Commits Pushes Branches Tags Pull Requests Pull Request Search Fork Clone

258264 ACTIVE Removing exposure control for test result flakiness in Test Management

Shantanu Shukla users/shshukla/removeflakyFF into master All resolved Approve Set auto-complete

Overview Files Updates Commits Conflicts

Policies

Required

- 1 reviewer approved
- Build in progress
- CredScan Validation succeeded

Optional

- All comments resolved
- Artifact Services Integration tests not run
- Packaging & Signing build not run
- Tfs.Self-Host Set 1 succeeded
- Tfs.Self-Host Set 2 not run

Status

- Doc publishing in progress
- Tfs.SelfTest - VSO.PR not queued
- TfsOnPrem.SelfTest - VSO.PR not queued
- Tfs.Deploy - VSO.PR has expired
- TfsOnPrem.SelfHost - VSO.PR not queu...

Work Items

No related work items

Reviewers

- EPS via Gaurav Sisodia
- Gaurav Sisodia Approved
- TCM

Description

Removing exposure control for test result flakiness in Test Management & Reporting, so that it is available to all scale units/accounts
Removed feature flag controlling availability of test reporting flakiness feature
Added associated unit test for UpdateTestResults API

Show everything

Add a comment...

Approved by Gaurav Sisodia 9/13/2017

C# TeamFoundationTestManagementResultService.cs 2 days ago

```

1855 1855      {
1856 1856          var fieldsNotFound = new List<string>();
1857 1857          foreach (var field in additionalFields)
1858 1858      {
1859          // do not populate OutcomeConfidence if TriFlakiness is turned off
1860          if(string.Equals(field.FieldName, TestExtensibilityConstants.OutcomeConfidence, StringComparison.OrdinalIgnoreCase) && !tfsRequestContext.IsFeatureEnabled(TestManagementFeatureFlags.TriFlakiness))
1861      {

```

Gaurav Sisodia 9/13/2017 Resolved
Please validate Tfs.deploy and Tfs.SelfHost for this change.

Shantanu Shukla 9/13/2017
Done.

Write a reply... Reactivate

C# TeamFoundationTestManagementTestResultServiceTests.cs 2 days ago

```

/Tfs/Service/TestManagement/Server.L0.Tests/TestResultService/TeamFoundationTestManagementTestResultServiceTests.cs

```

Perf Feedback | Resources | Scripts: 4/4 (6.2 MB) | CSS: 3/3 (1.3 MB) | Ajax: 1/5 | Other: 19/20 | Performance | TTI: 1921ms | SQL: 59 | Diagnose Bundles

Policies to keep master branch healthy (green)

- Required reviewers
- Build must pass
- Security plugins

(opt-in) Run functional tests in the cloud

VSONline / VSONline Te... | Dashboards | Code | Work | Build and Release | Test | ... | Search this project

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

Build VSO.PR_20170915.454

- Build
 - Initialize Job
 - Get sources
 - ChangImpactAnalysis
 - Init
 - Pre-Scorch
 - Build solution dirs.proj
 - Validate Build
 - Check for Warnings
 - Ensure REST Clients up to date
 - Run L0 Tests
 - Check for Test Warnings
 - Publish L0 Test Results
 - Cleanup Last Run Results from Contai...
 - Deploy L1 scripts to Container Direct...
 - Vssi VsTestConsole to Container
 - Vssi MsTestV2 to Container
 - Vssi L1.Sql.Tests to Container
 - Fetch Docker User Credentials
 - Login to docker repository
 - Download docker image
 - Run L1 Tests in container
 - Publish L1 Test Results from Container
 - Clean up suspended processes
 - Stop & Delete Container If Exists
 - Create DeployTools
 - Update Artifact

VSO.PR / Build VSO.PR_20170915.454

Edit build definition | Queue new build... | Download all logs as zip | Send Email | Retain indefinitely | Release

Build succeeded

Build VSO.PR_20170915.454 100%
Ran for 27.3 minutes (TfsAgentPool), completed 12.9 minutes ago

Summary | Timeline | Artifacts | Code coverage* | Tests | Build Target | WhiteSource Bolt Build Report

Build details

Definition	VSO.PR (edit)
Source	258264
Source version	Commit 31e6a305
Requested by	Microsoft.VisualStudio.Services.TFS on behalf of Shantanu Shukla
Queue name	TfsAgentPool
Queued	Friday, September 15, 2017 11:40 PM
Started	Friday, September 15, 2017 11:40 PM
Finished	Saturday, September 16, 2017 12:08 AM
Retained state	Build not retained

Associated changes

d3542e4 Authored by Shantanu Shukla
Removing exposure control for test result flakiness

Work items linked to associated changes

No work items linked to associated changes found for this build

Docs

Test Results

Total tests: 60286 (+60286)

- Passed (60286)
- Failed (0)
- Others (0)

Failed tests: 0 (+0)

- New (0)
- Existing (0)

Pass percentage: 100% (+100%)

Run duration: 6m 49s (+6m 49s)

Not Reported: 269

Detailed report >

Code Coverage

No build code coverage data available.

Tags

Add tag...

Deployments

Environments	Status
Aex.SelfTest - V... / no release created	⚠ Tags set for release trigger did not match the tags in build branch
CIX.SelfTest / no release created	⚠ Tags set for release trigger did not match the tags in build branch
Commerce.Self... / no release created	⚠ Tags set for release trigger did not match the tags in build branch

Fast and reliable signals

All unit tests (L0/L1) run in PR



VSOnline / VSOnline Te... | Dashboards | Code | Work | Build and Release | Test | ... | Search this project

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

Sps.SelfHost - VSO.CI / VSO.CI_20170913.76.1

Summary | Environments | Artifacts | Variables | General | Commits | Work items | Tests | Logs | History

Deploy | Save | Abandon | Send Email

Total tests: 1228 (Passed: 1228, Failed: 0, Others: 0) | Pass percentage: 100% | Run duration: 1h 5m | Not Reported: 77

Test	Failing since	Failing release	Duration
421/440 Passed - Sps.Service.SelfH...			0:26:31.213
GroupLicensingRuleReparentHa...	2 days ago	Current release	0:00:41.410

GroupLicensingRuleReparentHandler_ExecuteA...
Failed on V-SIVASP21269A | Log
Duration 0:00:41.410, 2 days ago

Error message

Test method
CollectionReparenting.Sps.Service.L2.Tests.GroupLicensingRuleReparentHa
threw exception:
Microsoft.VisualStudio.Services.WebApi.VssServiceResponseException: TF4
4a6b-93ec-eb3e58acae5. ---> System.TimeoutException: TF400898: An Inte
eb3e58acae5.

Stack trace

```
at Microsoft.VisualStudio.Services.WebApi.VssHttpClientBase.<HandleResp  
--- End of stack trace from previous location where exception was throw  
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()  
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebug  
at Microsoft.VisualStudio.Services.WebApi.VssHttpClientBase.<SendAsync>  
--- End of stack trace from previous location where exception was throw  
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()  
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebug  
at Microsoft.VisualStudio.Services.WebApi.VssHttpClientBase.<SendAsync>  
--- End of stack trace from previous location where exception was throw  
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()  
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebug  
at Vssf.TestFramework.Client.L2.SqlNotificationHttpClient.<FlushAsync>d  
--- End of stack trace from previous location where exception was throw  
at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()  
at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebug  
at CollectionReparenting.Sps.Service.L2.Tests.ReparentCollectionTestBas  
at CollectionReparenting.Sps.Service.L2.Tests.GroupLicensingRuleReparen
```

Attachments (1) | Bugs (1)

Perf Feedback | Resources | Scripts: 19/19 (5.8 MB) | CSS: 7/7 (1.1 MB) | Ajax: 7/22 | Other: 10/22 | Performance | TTI: 3099ms | SQL: 6 | REST: 1 | Total Remote: 7 | Diagnose Bundles

CI runs functional (L2) test suites

Test reliability is actively managed

Tests are trusted

Bug Cap

Everyone follows a simple rule they call the “Bug Cap”:

$$\# \text{engineers on your team} \times 4 = ?$$



Bug Cap

Everyone follows a simple rule they call the "Bug Cap":

$$10 \times 4 = 40$$

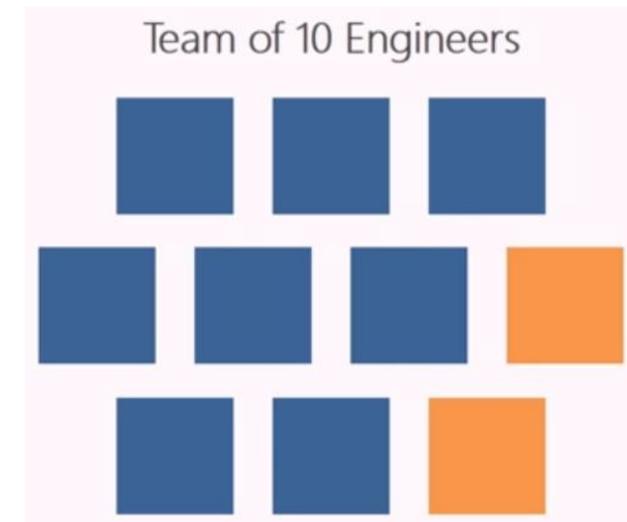
Rule: If your bug count exceeds your bug cap... stop working on new features until you're back under the cap.



Shielding Distractions: L-Team

A strategy adopted by their teams to provide focus, and assist with an interrupt culture

The team self-organizes each sprint into two distinct sub-teams: Features and Shield



Rotating cadence established by the team



VSTS Scorecard

Engineering Scorecard - Sprint 109

Help

		# LSI repair WIs older than 2 sprints	Livesite Health % 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	Engineering Debt # of Security WIs older than 21 days	DevFabric L2 SelfTest & SelfHost Reliability	# of Cloud TRA Tests
Agile	Blueprint	0	N/A	0	0	1	5	0	100 (2/2)	0
	Modern WIT	0	100	1	N/A	0	2	0	100 (66/66)	244
	WIT Workflows	0	N/A					0	100 (47/47)	0
	WIT PI	0	N/A					0	100 (11/11)	0
	WIT IQ	0	100					0	100 (60/60)	0
enSPIRe	Keros	0	N/A					0	N/A	0
	Subscriptions	1	75					0	N/A	0
	Engineering Productivity	0	N/A					1	N/A	0
	Enterprise Requirements	0	N/A					0	N/A	0
	Service Insights	0	100					0	N/A	0
Enterprise Sharing	Gift	0	100					0	N/A	0
	OSS	0	100					0	N/A	0
NC DevX	Build	0	100	1	33	5		0	91 (21/23)	0
	VC Workflows	0	N/A					0	100 (10/10)	0
	Java	0	N/A					0	100 (11/11)	0
	VCX Mavericks	0	N/A					0	N/A	0
	VC Core Experience	0	N/A	2		11		0	75 (9/12)	26
NC Services Platform	VC First Party	0	N/A					0	100 (36/36)	46
	VCX Agile	0	N/A					0	N/A	0
	Open ALM	1	66					1	75 (9/12)	50
	Cloud Admin and Tools	0	100					4	90 (9/10)	153
	Cloud Services Framework	0	100					8	97 (142/146)	428
Reporting	Enterprise Analytics	0	100					0	N/A	0
	Dashboards	0	N/A					0	100 (60/60)	0
Service Delivery	Service Delivery	8	75					6	N/A	0
	Identity	12	45	4		6		2	100 (62/62)	229
Shared Cloud Services	Enterprise Authorization	0	100	1		6		1	99 (112/113)	49
	User Engagement	0	N/A					0	100 (19/19)	0
	Commerce and Licensing	1	85	1		8		0	92 (11/12)	628
VSO	VSCS	27	75	21		10	5	23	N/A	0

Things they don't watch

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



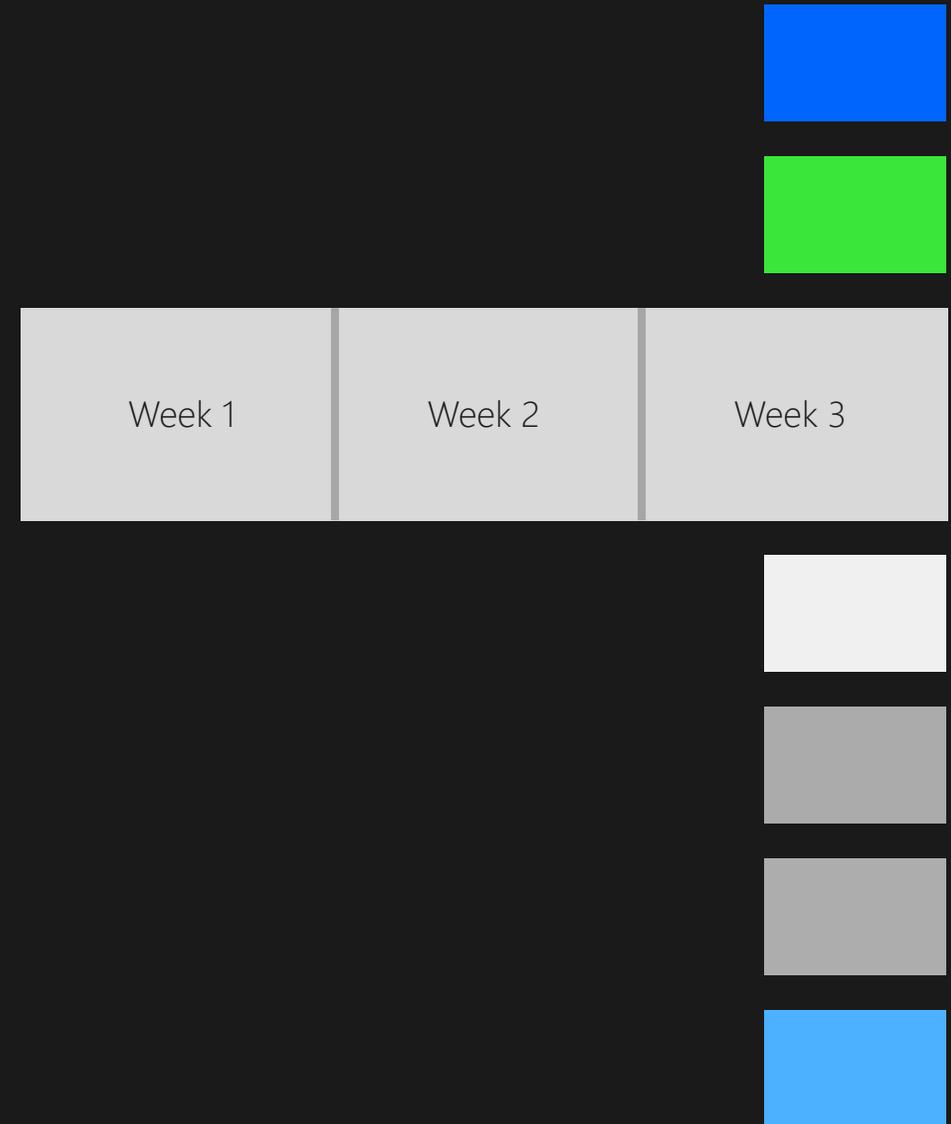
Day in the life of an Engineer

Work in Master

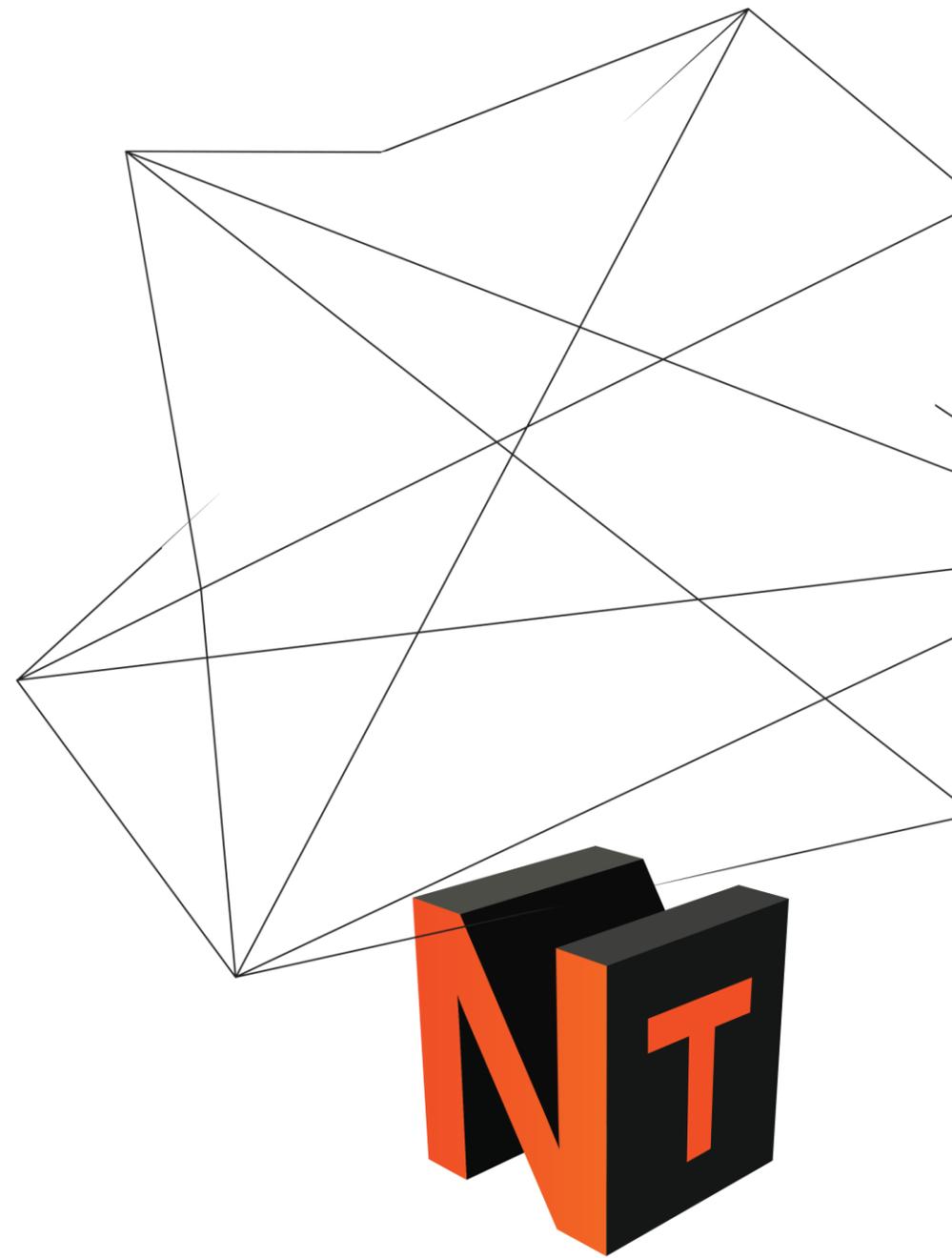
Walk the walk

Continuous deployment

Instrument everything

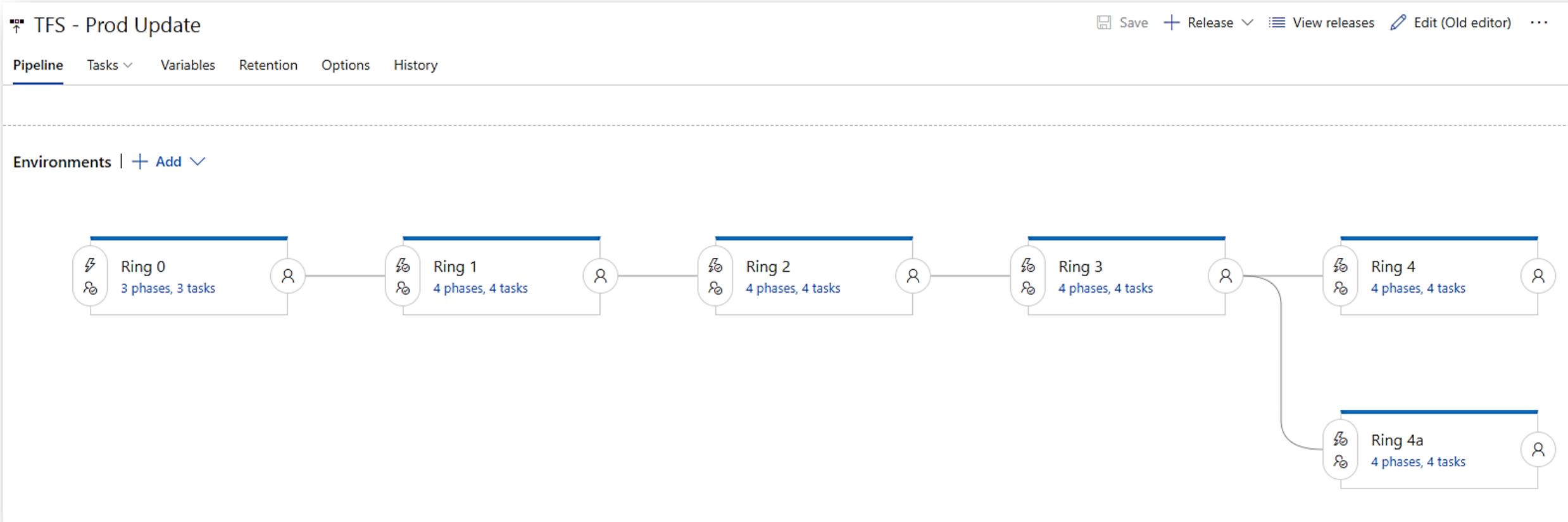


Release Management



#ntk18

RM Deployment Definition and Exposure Control



Rings are modeled as Environments

Different rings serve different customer groups

NT Rings provide control over customer exposure to features (and bugs) #FTK18

Ring Definition

TFS - Prod Update

Pipeline Tasks Variables Retention Options History

Ring 1

Deployment process

Run on server

Run on server

Pause Between Rings
Manual Intervention

Delays are achieved via manual intervention tasks configured to continue after the delay

Run on agent

Run on agent

Update Binaries
Vssf.ProdUpdate

Run on server

Run on server

Pause for AT DT mode
Manual Intervention

Run on agent

Run on agent

Update Database
Vssf.ProdUpdate

Vssf.ProdUpdate ⓘ

Version

1.*

Display name *

Update Binaries

ServiceName * ⓘ

\$(ServiceName)

ProductUri * ⓘ

\$(ProductUri)

Ring * ⓘ

\$(Ring)

DeployCommand * ⓘ

\$(UpdateBinariesCommand)

ProductGroups * ⓘ

\$(ProductGroups)

Control Options

All the heavy lifting is done by this custom command line tool, which is the same tool they use to deploy in dev and test environments

#ntk18



Automated Rollback

Monitor perf counters after a deployment

If 4 of 9 samples exceed threshold, rollback

Failures automatically collect a reflected dump and upload it

Once data-tier servicing happens we only move forward (no rollback)

Counter	Threshold
ASP.NET v4.0.30319\Requests Queued	25
Logical Disk\% Free Space	10
Memory\Available Mbytes	256
Processor(_Total)\% Processor Time	95%
TFS Services:Service Bus(_Total)\Client Notifications Average Send Time	10s
TFS Services\Average SQL Connect Time	3s
TFS Services\Active Application Service Hosts	4000



Quality and Release Management

Practices to safeguard Production

- Safe deployment practices

- Exposure control

- A/B Testing

Telemetry and monitoring

- Failures, exceptions, performance, security ...

Simulating failures

- Fault injection and Chaos Engineering

- Failover testing

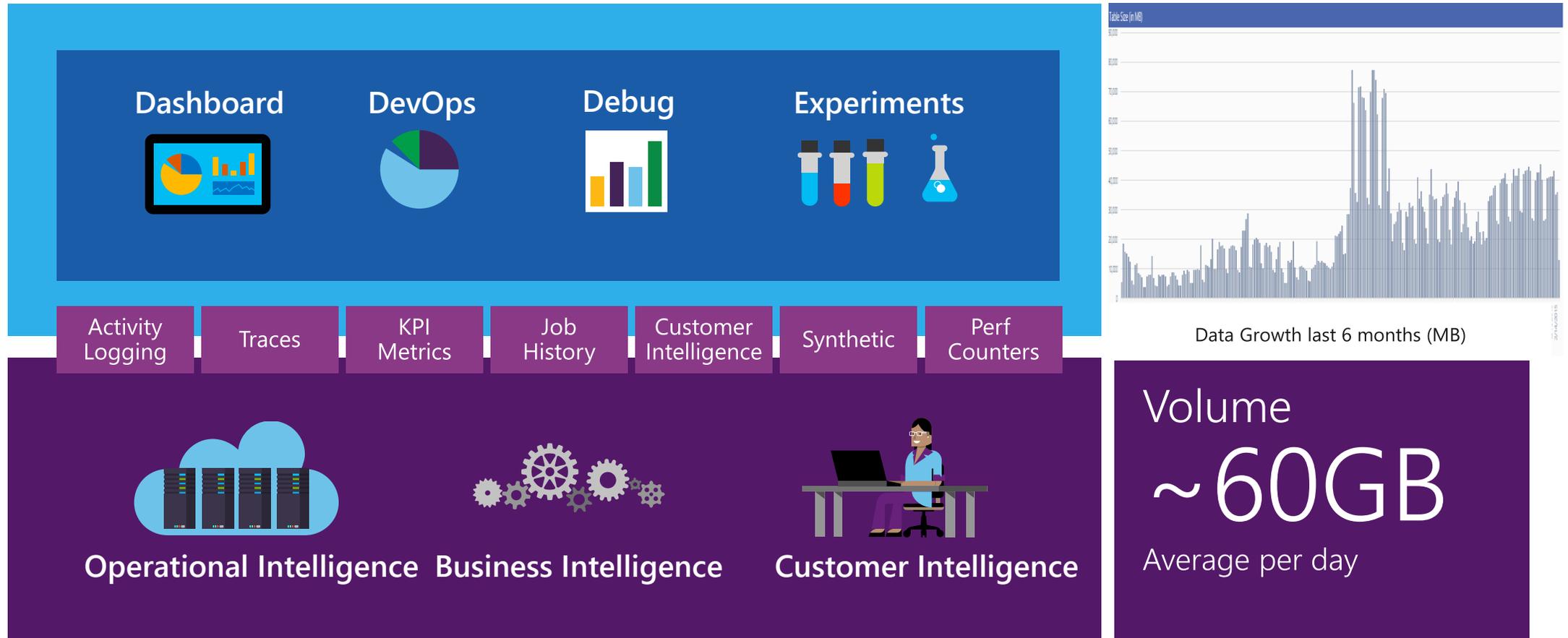
Yes, we run tests in production!

- L3 tests against production

- Pre-flight and Ring 0 environments

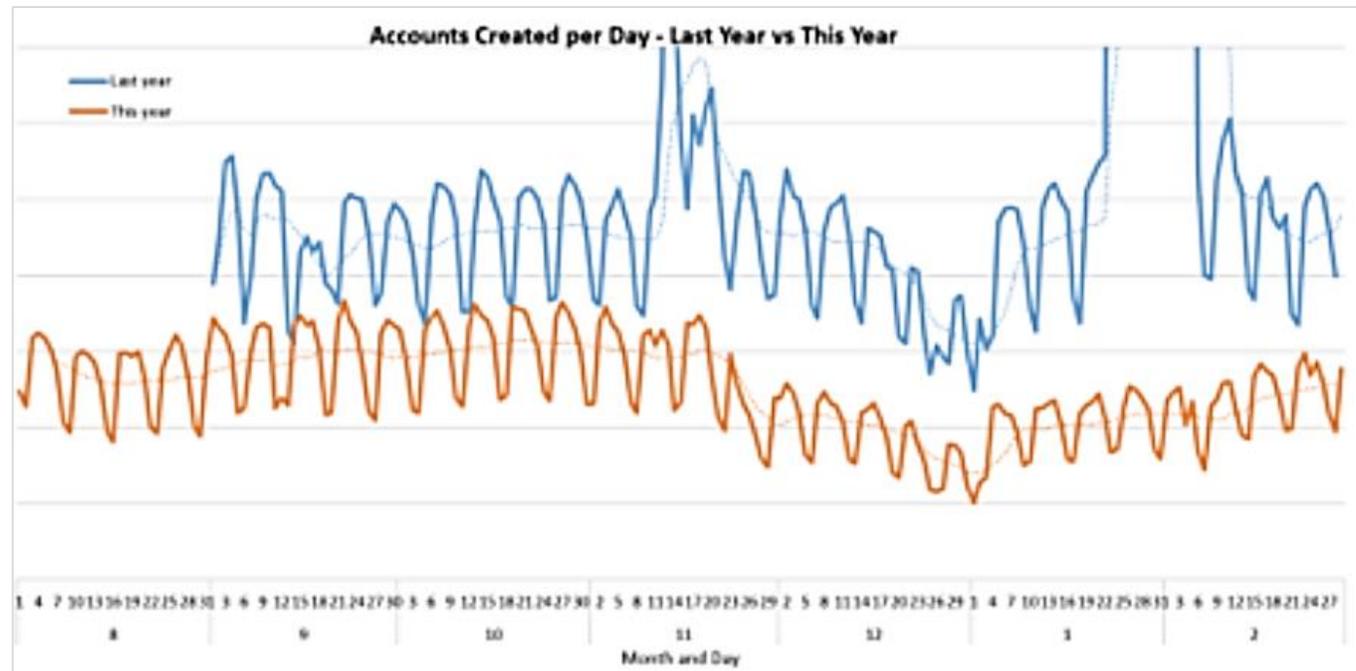


Gather everything

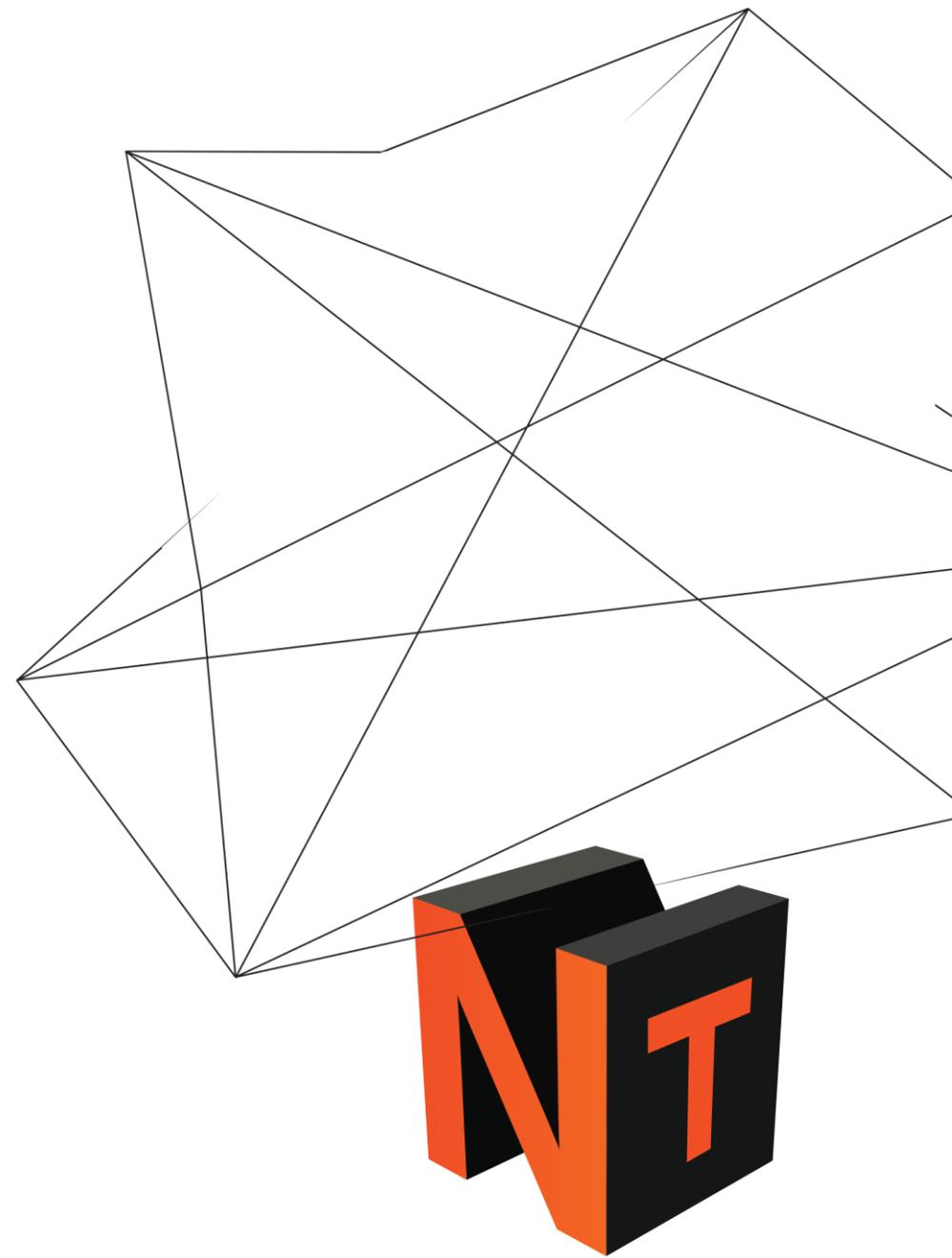


Running the Business on Metrics

- Start with what is most important/painful & evolve
- Designing metrics is as hard as designing features
- Bake it into the review culture to spur activity

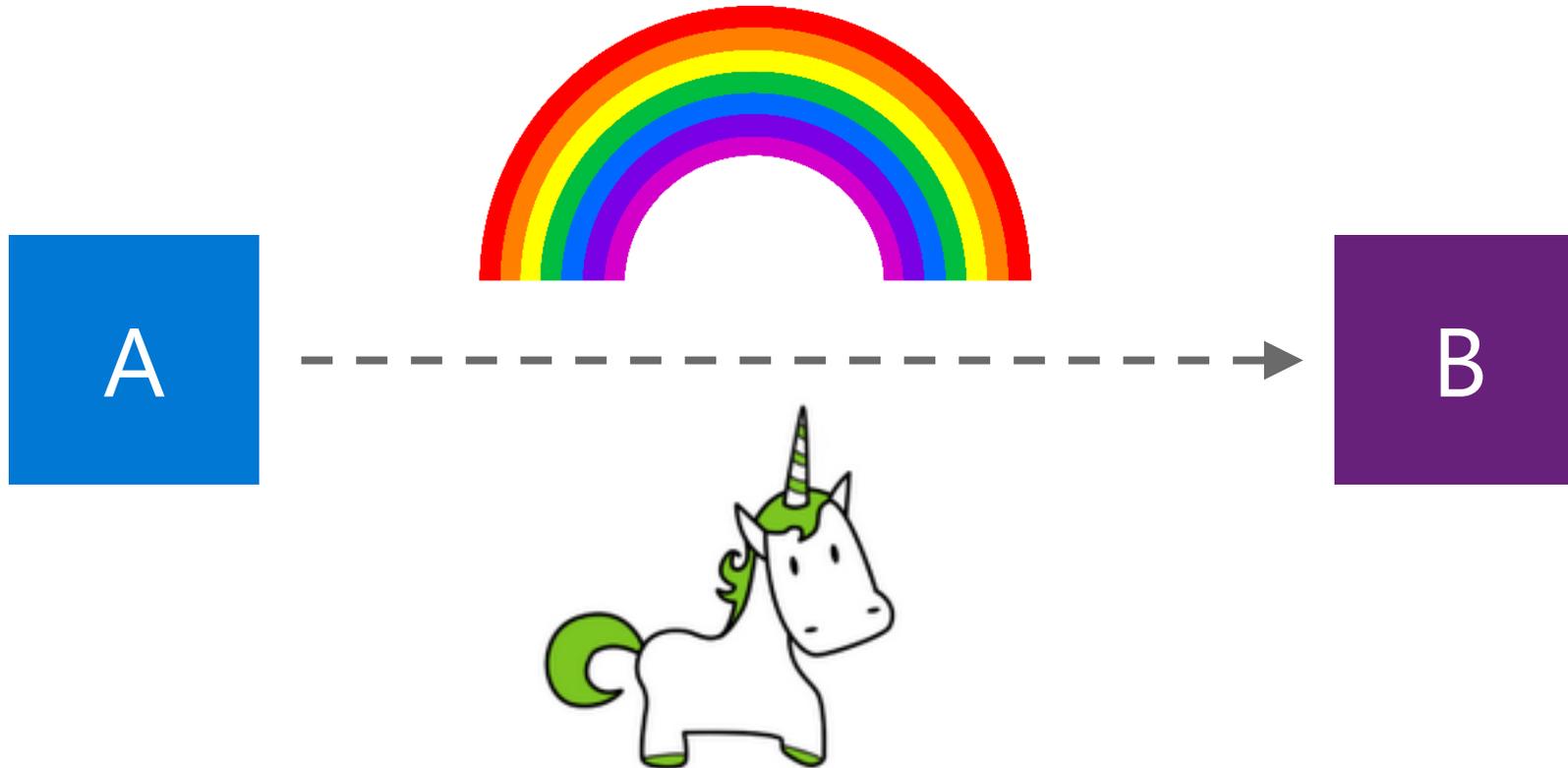


Too good to be true?

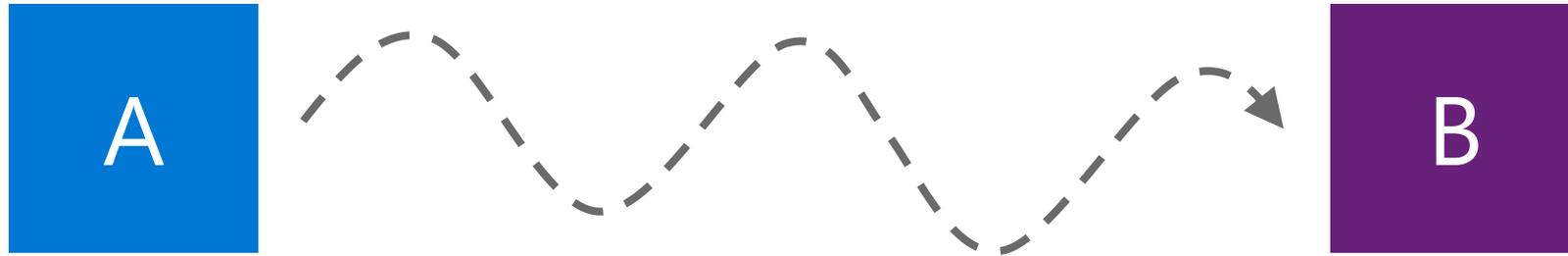


#ntk18

Too good to be true?



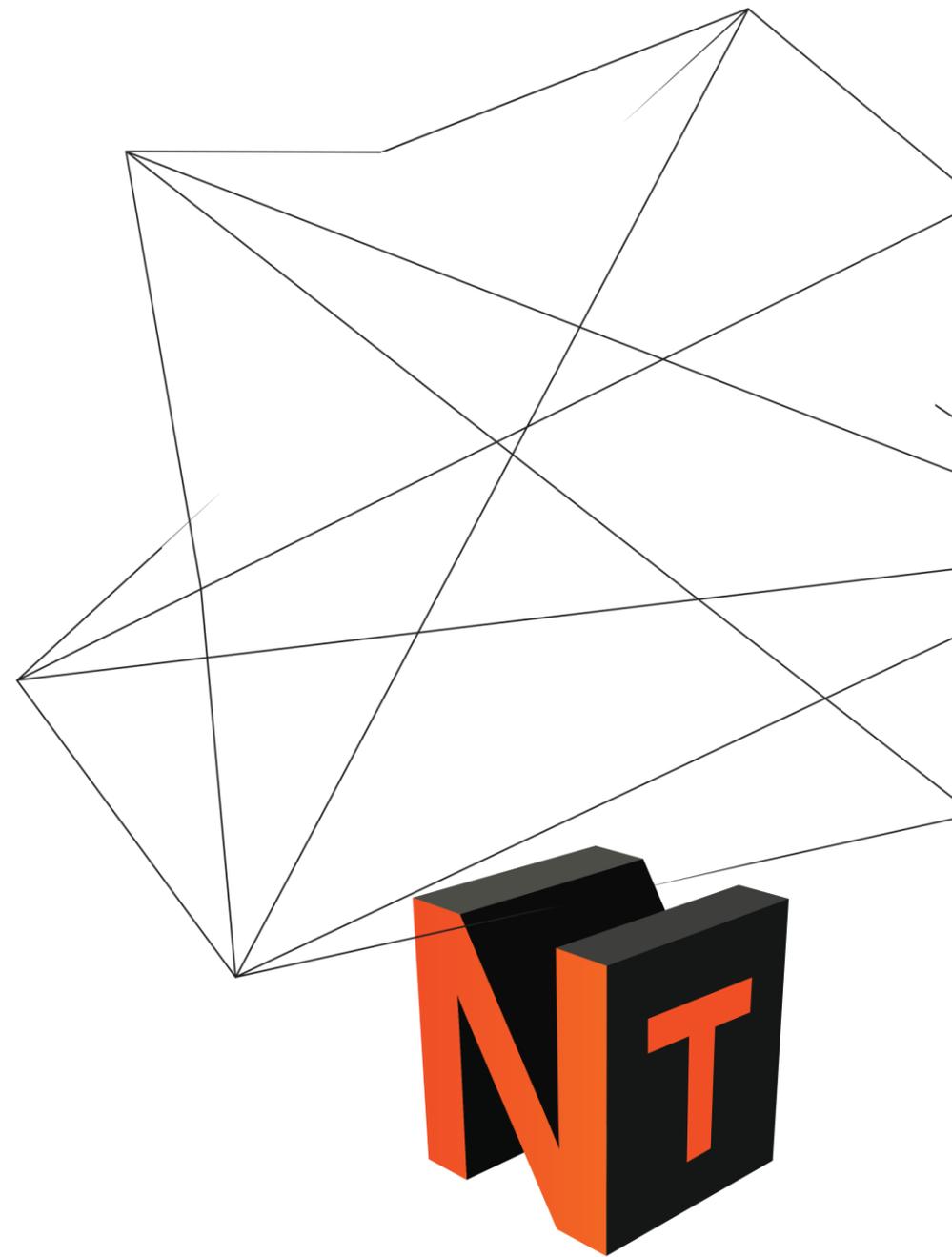
... a journey of continued improvement.



This journey does not end.



Is it really working?



#ntk18

Transformation

Before

- 4-6 month milestones
- Horizontal teams
- Personal offices
- Long planning cycles
- PM, Dev, Test
- Yearly customer engagement
- Feature branches
- 20+ person teams
- Secret roadmap
- Bug debt
- 100 page spec documents
- Private repositories
- Deep organizational hierarchy
- Success is a measure of install numbers
- Features shipped once a year

After

- 3-week sprints
- Vertical teams
- Team rooms
- Continual Planning & Learning
- PM & Engineering
- Continual customer engagement
- Everyone in master
- 8-12 person teams
- Publicly shared roadmap
- Zero debt
- Specs in PPT
- Open source
- Flattened organization hierarchy
- User satisfaction determines success
- Features shipped every sprint



Transformation Benefits

Teams feel that they own the customer experience & are responsible for improving it

Teams are continually planning

Planning is driven by continual learning

- Telemetry on usage

- Customer feedback

- "Failing fast" through incremental execution and delivery

Opportunities to continually evaluate progress

We can react... if & when we need to change course



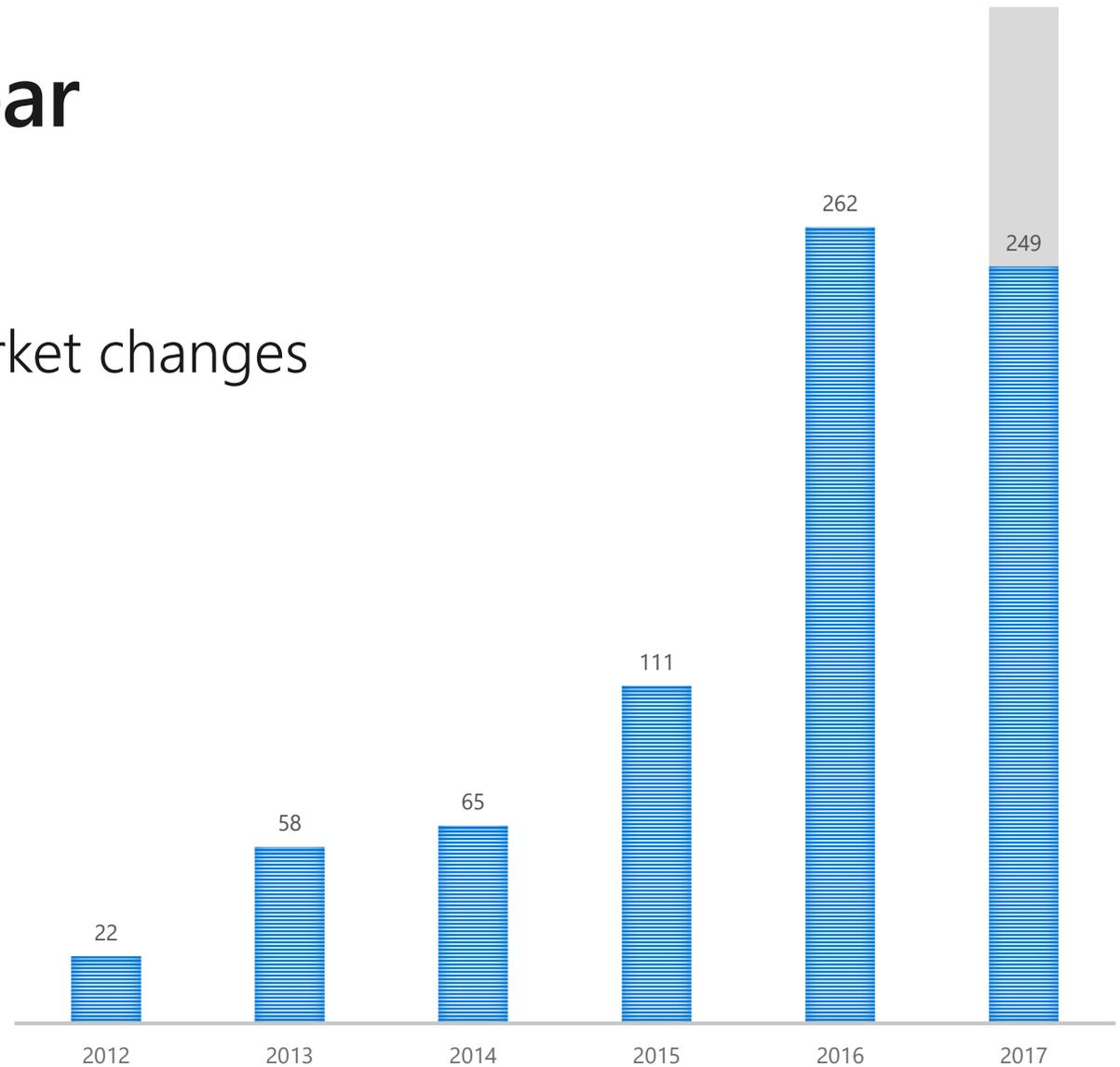
Features Delivered per Year

Deliver more value to customers

Faster responses to customers and market changes

Improved engineering satisfaction

2x productivity increase



Thank you!



Ognjen Bajić
obajic@ekobit.hr



Ana Roje Ivančić
aroje@ekobit.hr