

# WorkshopPLUS: DevOps Fundamentals

Module 0: Business Value



# Conditions and Terms of Use

## Microsoft Confidential

This training package is proprietary and confidential and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet website references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

## Copyright and Trademarks

© 2016 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Use of Microsoft Copyrighted Content** at  
<https://www.microsoft.com/en-us/legal/intellectualproperty/permissions/default.aspx>

Microsoft®, Internet Explorer®, Outlook®, SkyDrive®, Windows Vista®, Zune®, Xbox 360®, DirectX®, Windows Server® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

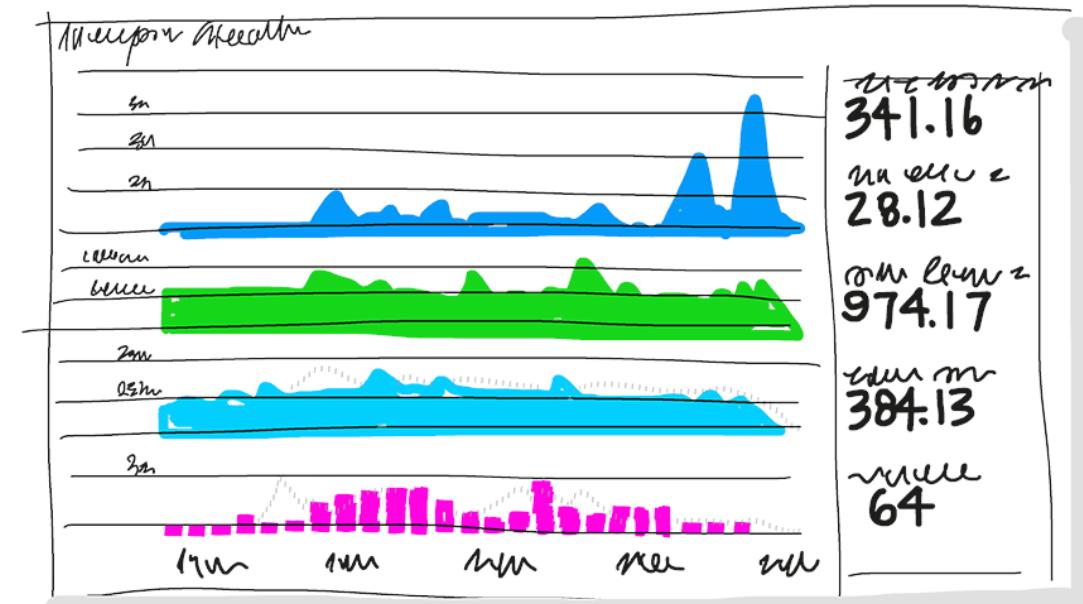
# Course Objectives

- Over the next four days, we will develop a microservice REST API using fundamental DevOps practices, Agile processes and procedures:
  - Agile at Scale (Lightweight Scaled Agile Framework Implementation)
  - Shift-Left Testing
  - Test-Driven Development (TDD)
  - Designing for Testability
  - Source Control Strategy (Using the Topic Branch Methodology)
  - Feature Flag Development
  - Continuous Integration Pipeline Development
  - Continuous Delivery Pipeline Development
  - Application Monitoring with Application Insights

# Module Objectives

After completing this module, you will:

- Understand the value proposition of DevOps,
- learn about potential organization changes, and
- see how the Microsoft Visual Studio Team (now Azure DevOps Team) transformed from Waterfall to DevOps.



# Digital Disruption is Happening

"Silicon Valley is coming. We are going to work hard to make our services as seamless and competitive as theirs."

Jamie Dimon  
*CEO, JP Morgan*

J.P.Morgan

"We are putting 70% of our applications in the cloud to improve flexibility. And we are launching truly game-changing applications to improve efficiency."

Jeffrey Immelt  
*Former CEO, GE*



"Walmart is re-defining the next generation of retail growth and is the best-positioned retailer to win at the convergence of digital and physical retail."

Doug McMillon  
*CEO, Walmart*

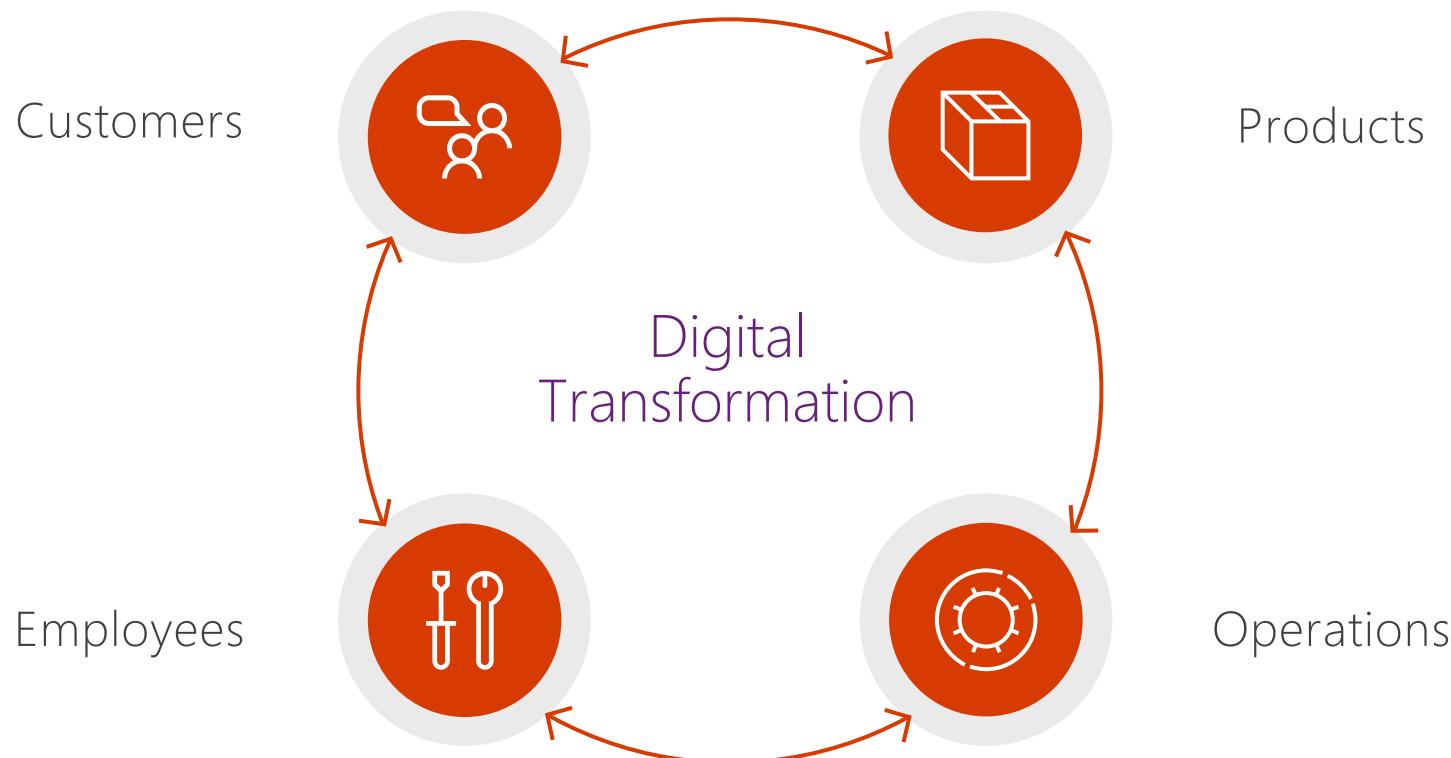
Walmart\*

"It's about looking ahead to the changes coming to the automobile business, particularly from potential 'disruptors' in Silicon Valley – and preparing Ford to thrive through those changes."

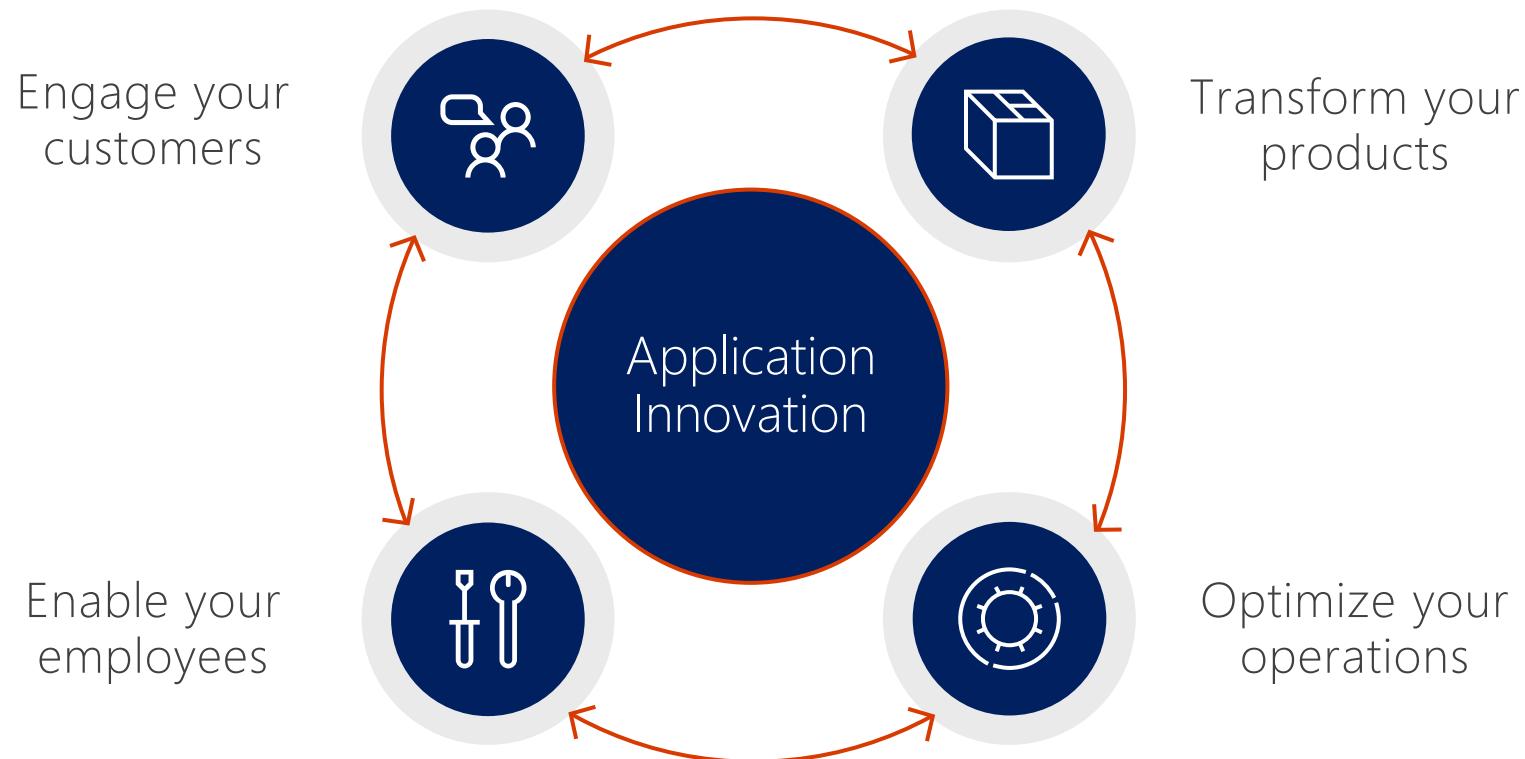
Mark Fields  
*Former CEO, Ford Motor Company*



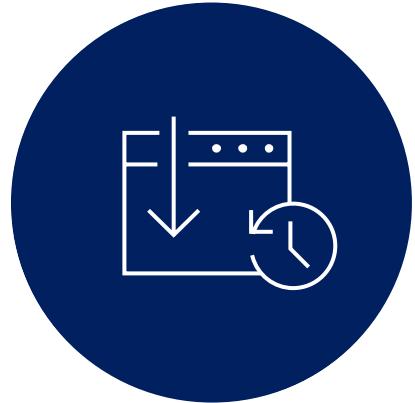
# Companies are Re-Aligning their Business



# Digital Transformation = Application Innovation



# DevOps = Application Innovation Enabler



Shorten cycle times  
and deliver value faster

Speed up development cycles  
with developer productivity tools

Automate the workflow between  
Dev and Ops and deploy faster

Continuously measure, learn and  
improve with state-of-the-art DevOps  
tools and an agile cloud platform

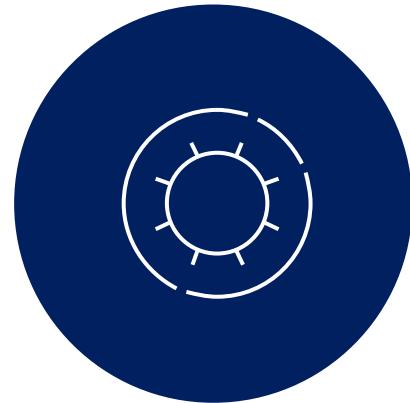


Improve quality and availability of  
apps and services

Easily plan, execute and monitor your  
entire QA effort, continuously

Fix issues faster with insights and  
information captured in production

Use cloud-based load testing to  
evaluate and validate application  
performance before you deploy



Optimize IT resources  
and eliminate waste

Improve developer agility and  
resource efficiency with cloud dev  
and test environments

Eliminate over-provisioning,  
lower costs and automate your  
deployment workflow with cloud  
environments for pre-production

# Application Innovation Trends

Technology	 <p>Multi-platform Multi-device</p>	 <p>Cloud services</p>	 <p>User-centric</p>	 <p>Social</p>
People & Processes	 <p>Continuous feedback</p>	 <p>Continuous quality</p>	 <p>Continuous delivery</p>	 <p>Heterogeneous development teams</p>

# Opportunity

60%

apps get to market late

Arriving to market too late

70%

apps go to market missing functionality

Poor quality

5

months on average to create a new app. Only 28% of companies worldwide create an app in less than 3 months

Lengthy lifecycles

Solution: Deliver Value Faster



## Testimonial #1: Industry Changes



**Brian Harry**  
Corporate Vice President

# Waterfall, Agile, and DevOps

## Waterfall

Project centric  
Long planning cycle  
The plan drives everything  
Siloed delivery phases  
No room for feedback within the cycle

## Agile

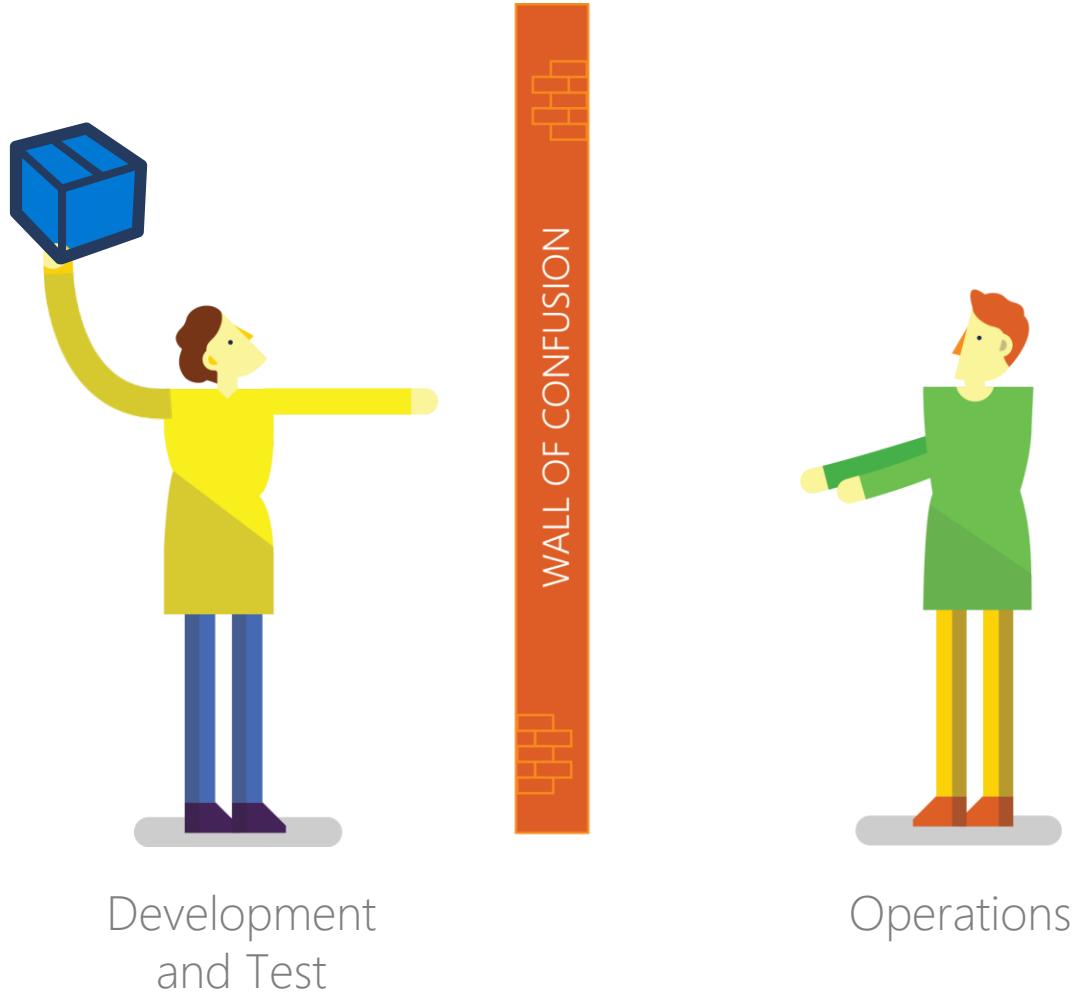
Software development centric  
Short planning cycles  
Designed for small software teams; however, can scale with the enterprise

## DevOps

Customer value centric  
Adopts agile planning and scales it  
Team of teams  
Cross functional teams  
Embraces validated learning

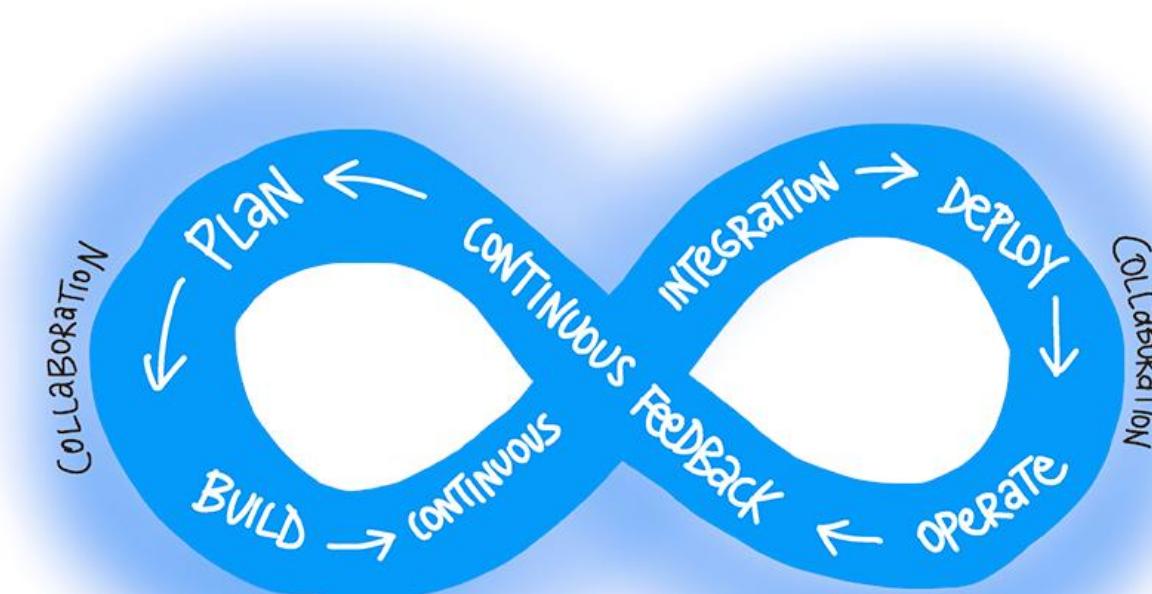
What is DevOps?

# Traditional Development and Operations



# Our Definition of DevOps

DevOps is the union of **people**, **process**, and **product** to enable continuous delivery of value to your customers. *Donovan Brown, Principle DevOps Manager, Microsoft*



# DevOps

- The contraction of “Dev” and “Ops” refers to [replacing siloed](#) Development and Operations to [create multidisciplinary teams](#) that now work together with [shared and efficient practices and tools](#).
- Essential DevOps practices include agile planning, continuous integration, continuous delivery, and monitoring of applications.

# Understanding DevOps Practices

"DevOps  
is using **Agile Planning**"

"DevOps is development  
and operations **working together**  
as a unified team"

"DevOps  
is using **automation**"

"DevOps  
is **small** deployments"

It's a boulder!

"DevOps is treating your  
infrastructure  
as code"

"DevOps  
is feature **switches**"

It's a branch!

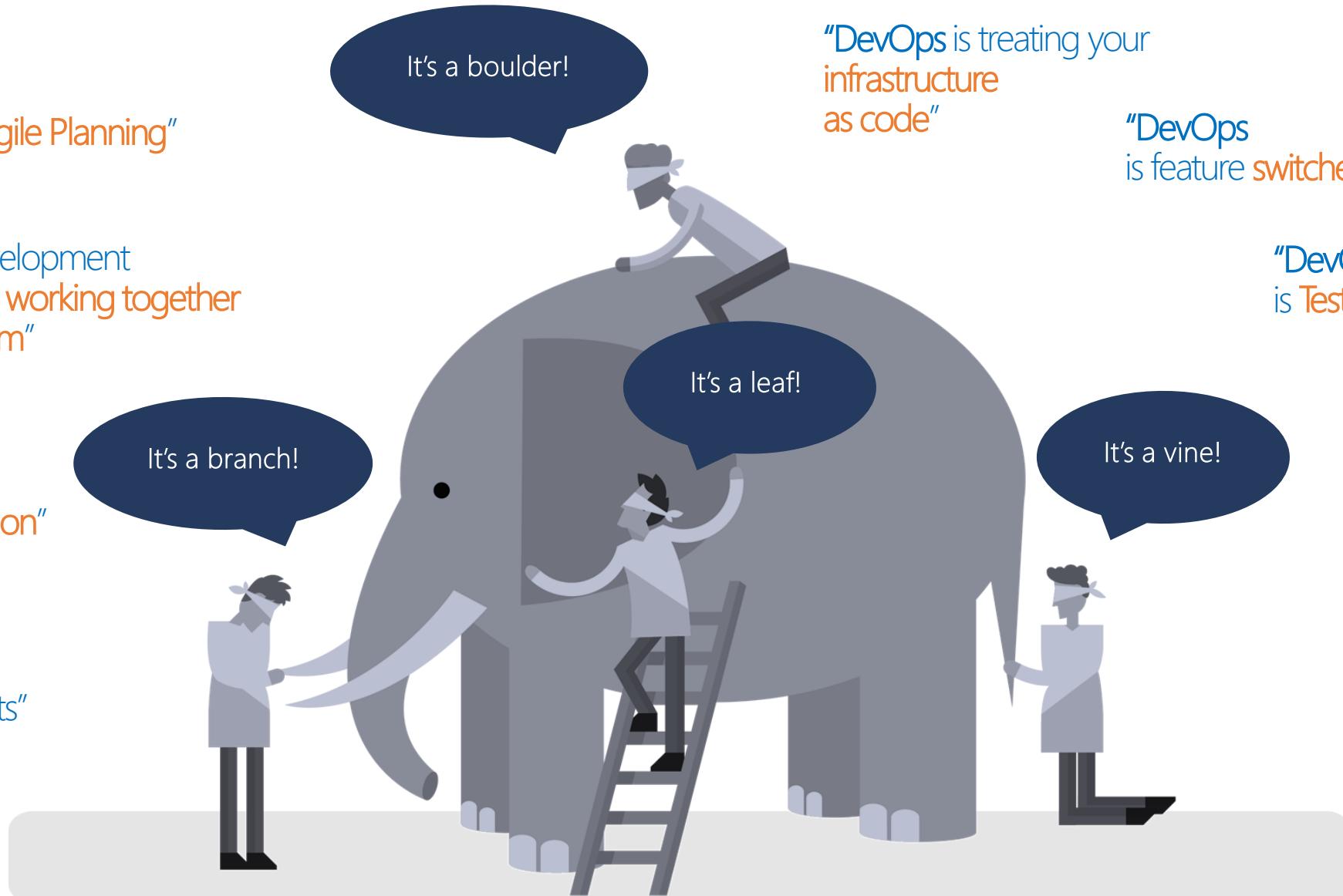
It's a leaf!

It's a vine!

"DevOps  
is **Testing in production**"

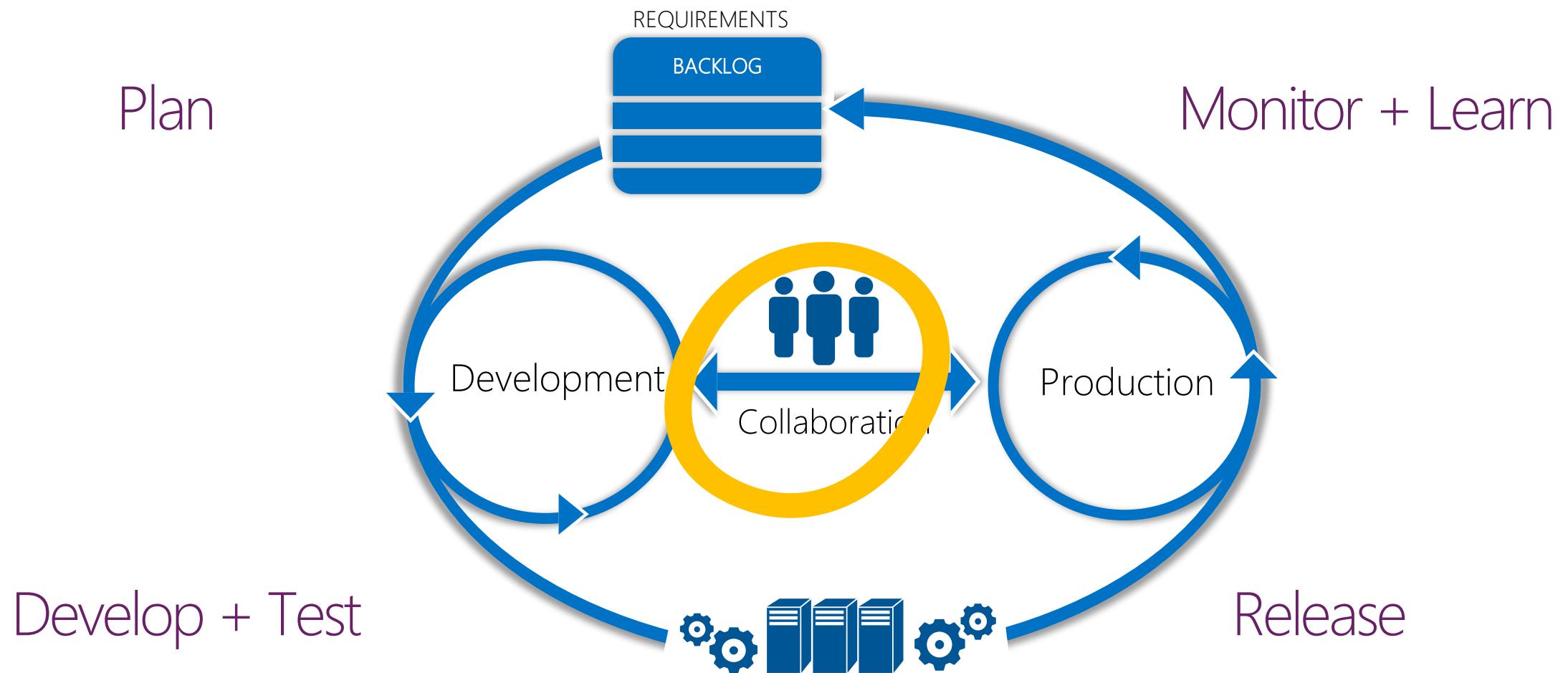
"DevOps  
is **A/B Testing**"

DevOps is  
"**Microservices**"



Why does your organization need DevOps?

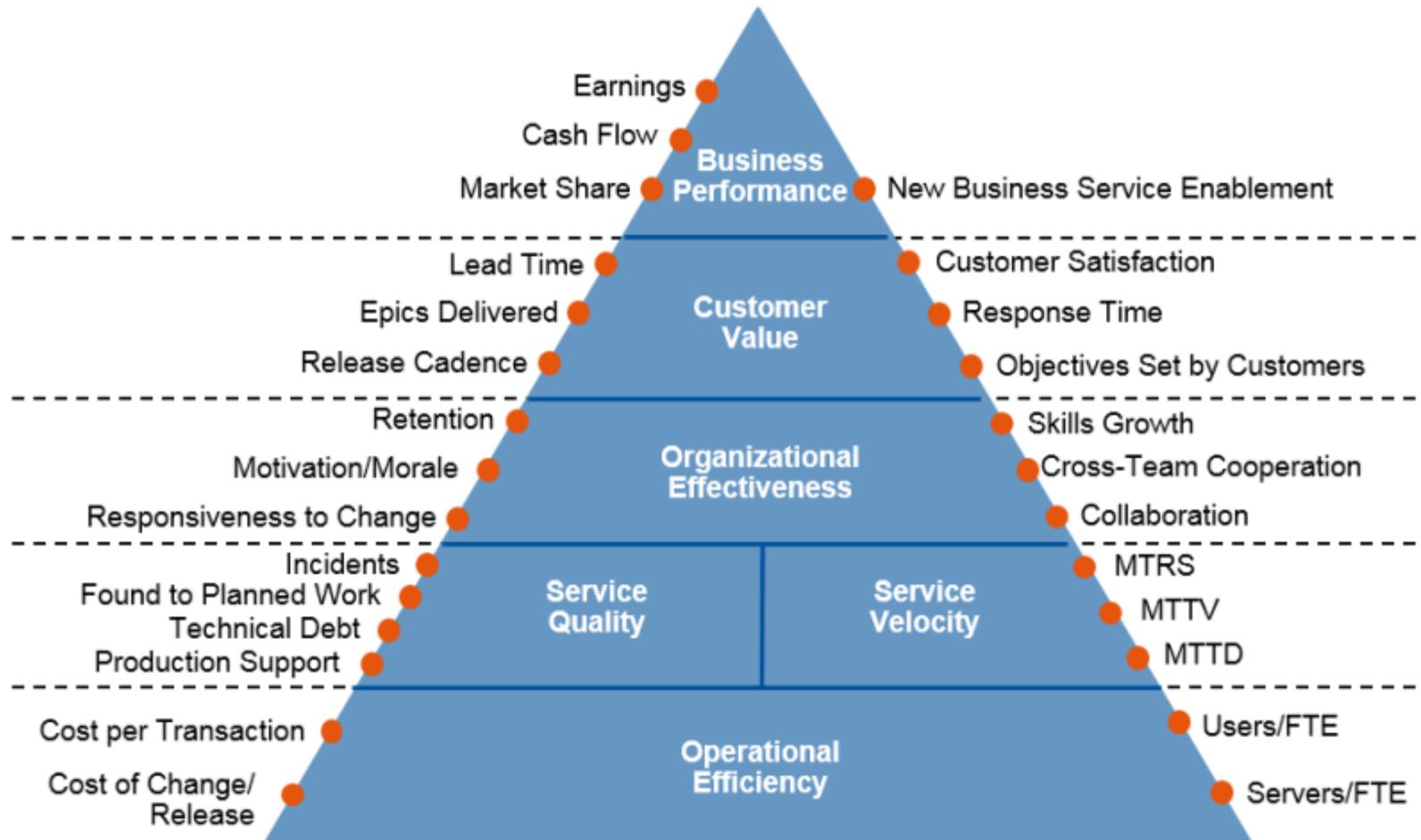
# Deliver Faster, Smarter, and Continuously



# Not Just a “Buzzword”

- Shorten your cycle time
- Stay competitive
- Produce value, quality, and secure features at a regular cadence
- Introduce collaboration among teams
- Remove the development chaos
- Implement predictable practices with repeatable and successful results
- Gain awareness of your software through monitoring

# DevOps Metrics



© 2017 Gartner, Inc.

# The Phoenix Project

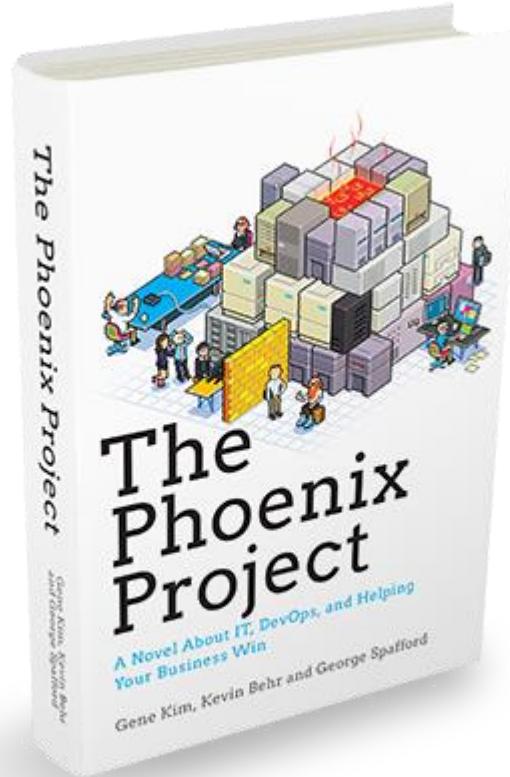
*The Phoenix Project* frames in the impact of chronic friction between Dev and IT Ops. This is a fantastic illustration of relatable scenarios that we all have experienced.

## The Three Ways

1. Maximize Flow
2. Amplify Feedback loops
3. Continual improvement

## Four Types of Work

1. Business Projects
2. Internal IT Projects
3. Changes
4. Unplanned / Recovery work



Bill is an IT manager at Parts Unlimited. It's Tuesday morning and on his drive into the office, Bill gets a call from the CEO.

The company's new IT initiative, code named Phoenix Project, is critical to the future of Parts Unlimited, but the project is massively over budget and very late. The CEO wants Bill to report directly to him and fix the mess in ninety days or else Bill's entire department will be outsourced.

# Competitive Advantage through Deployment

- Faster Deployments
- Higher Quality and Security
- Lower Lead Time
- Shorter Cadence

Company	Deploy Frequency	Lead Time	Reliability	Customer Responsiveness
Amazon	~23,000 / day	Minutes	High	High
Google	~5,500 / day	Minutes	High	High
Netflix	~500 / day	Minutes	High	High
Facebook	~1 / day	Hours	High	High
Twitter	~3 / week	Hours	High	High
Traditional Enterprise	~1 / 9 months	Months or Quarters	Low/ Med	Low/ Med

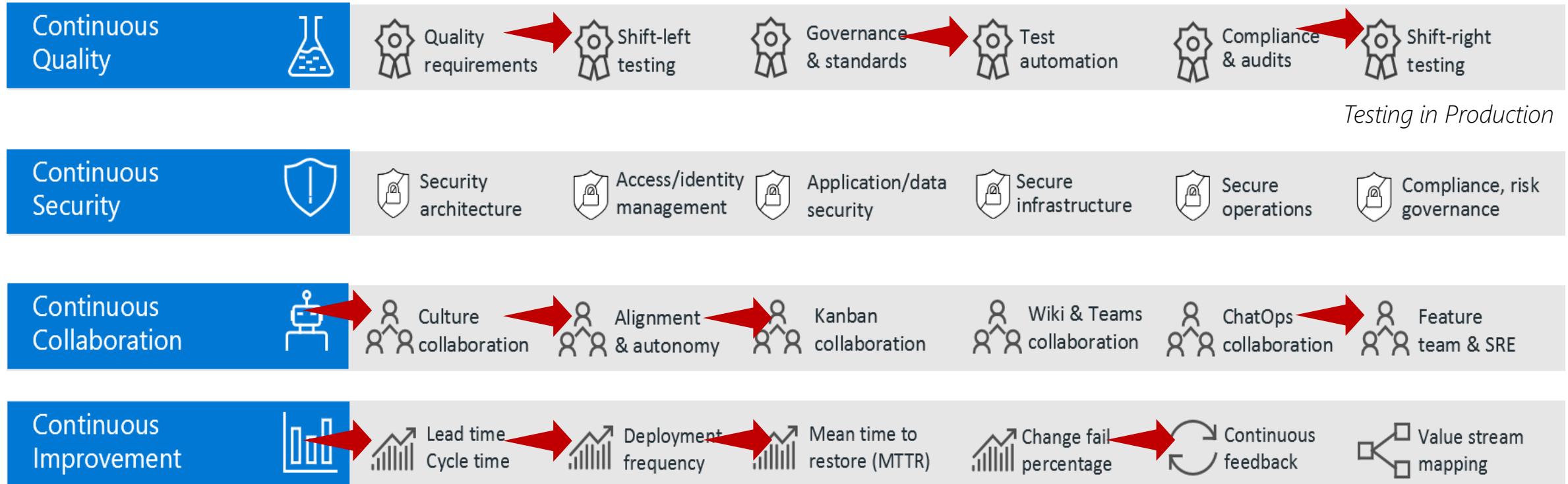
How does your organization start a DevOps  
**Communication**  
journey?

# DevOps Capabilities and Practices

Continuous Planning	Continuous Integration	Continuous Delivery	Continuous Operations
<ul style="list-style-type: none"><li>Objectives &amp; Key Results (OKR)</li><li>Lean Product discovery</li><li>Lean Product definition</li><li>Release planning</li><li>Sprint planning</li><li>Agile requirement</li><li>Security requirement</li><li>Architecture design</li><li>Capacity planning</li><li>UX architecture design</li><li>Threat modeling</li><li>Prioritization &amp; estimation</li><li>Demo &amp; retrospectives</li></ul>	<ul style="list-style-type: none"><li>Behavior-driven development (BDD)</li><li>Test-driven development(TDD)</li><li>Microservices &amp; container development</li><li>Multi-repo vs. Mono repo</li><li>Unit test/code coverage</li><li>Version control</li><li>GIT pull request</li><li>Trunk-based policies<ul style="list-style-type: none"><li>Security static code scan</li><li>CredScan</li><li>OSS component compliance</li></ul></li><li>Build pipeline – parallel/serial</li></ul>	<ul style="list-style-type: none"><li>Release pipeline</li><li>Secure infra deployment</li><li>IaaS deployment</li><li>PaaS deployment</li><li>SaaS deployment</li><li>Shared services</li><li>Infra-as-code</li><li>Change management</li><li>Configuration management</li><li>Release management</li><li>Blue/Green deployment</li><li>Canary deployment</li><li>Feature flag</li><li>Trunk always PRD ready</li></ul>	<ul style="list-style-type: none"><li>Site Reliability Engineering(SRE)</li><li>Telemetry/monitoring</li><li>Application perf. monitoring</li><li>Auto failover &amp; scaling &amp; DR</li><li>Modern service management</li><li>Secure access/app/data</li><li>High availability/security/cost and performance advisory</li><li>Secure Dev/Ops ChatOps</li><li>Shift-right testing</li><li>Secrets management</li><li>Governance</li><li>Automation &amp; AIOps</li><li>Continuity &amp; resilience</li></ul>

→ Indicates capabilities and practices discussed in this course, and some of these have labs directly associated with them

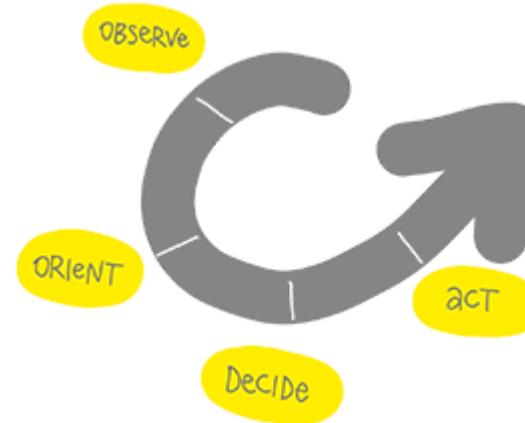
# DevOps Capabilities and Practices



Indicates capabilities and practices discussed in this course, and some of these have labs directly associated with them

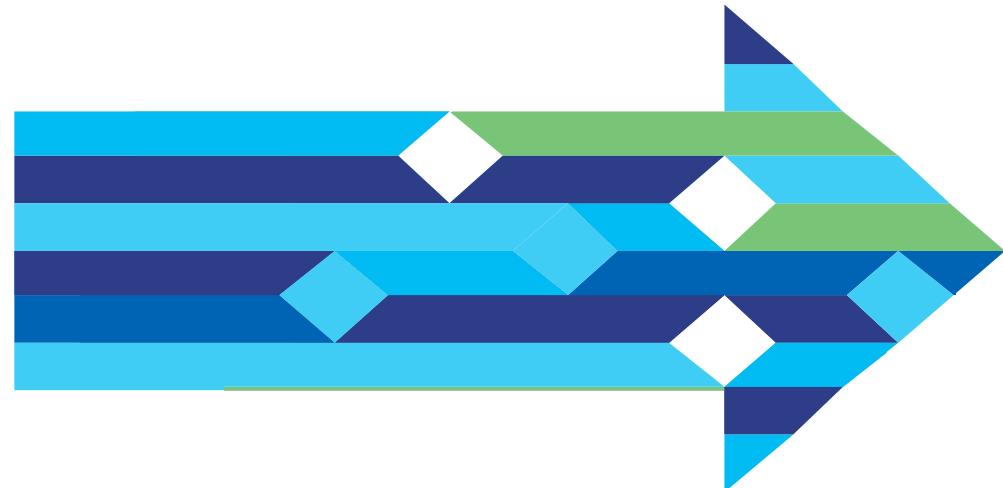
# Understand your Cycle Time with OODA

- Observe business, market, needs, user behavior, and telemetry data
- Orient with the enumeration of options for what you can deliver
- Decide what to pursue, and
- Act by delivering working software to real users



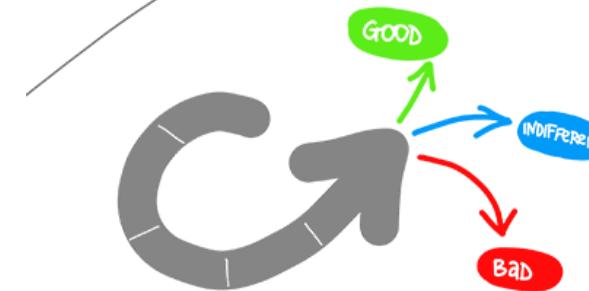
# Become Data Informed

- Use data to inform what you will do in your next cycle
- Experiment; Fail fast on those experiments that don't advance the business and double down on those that support the business



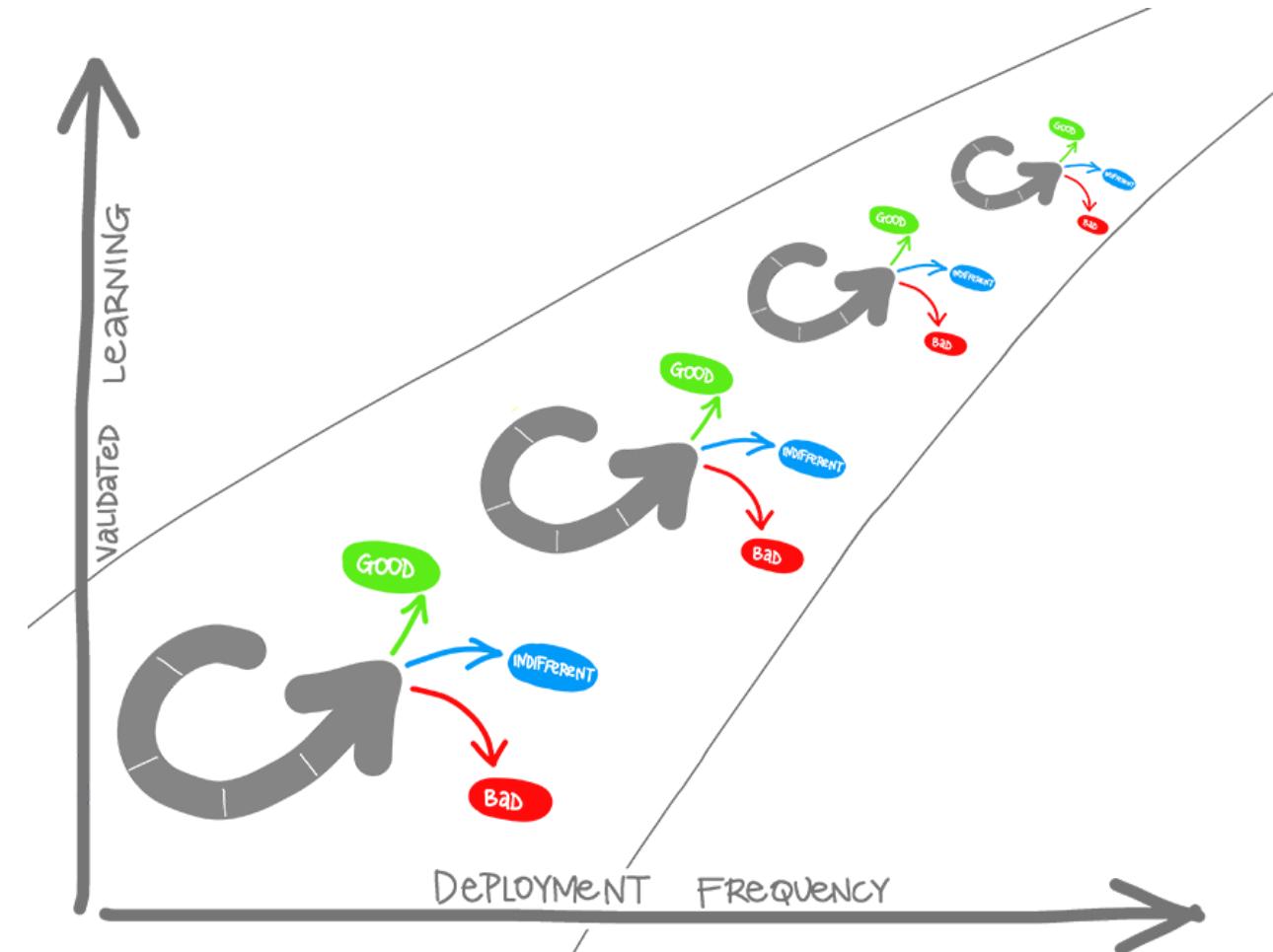
# Strive for Validated Learning

- How quickly you can fail fast or double down is determined by how long that loop takes, or in lean terms, by your cycle time
- Your cycle time determines how quickly you can *gather feedback* to determine what happens in the next loop
- The feedback that you gather with each cycle should be *real, actionable data*. This is called validated learning.



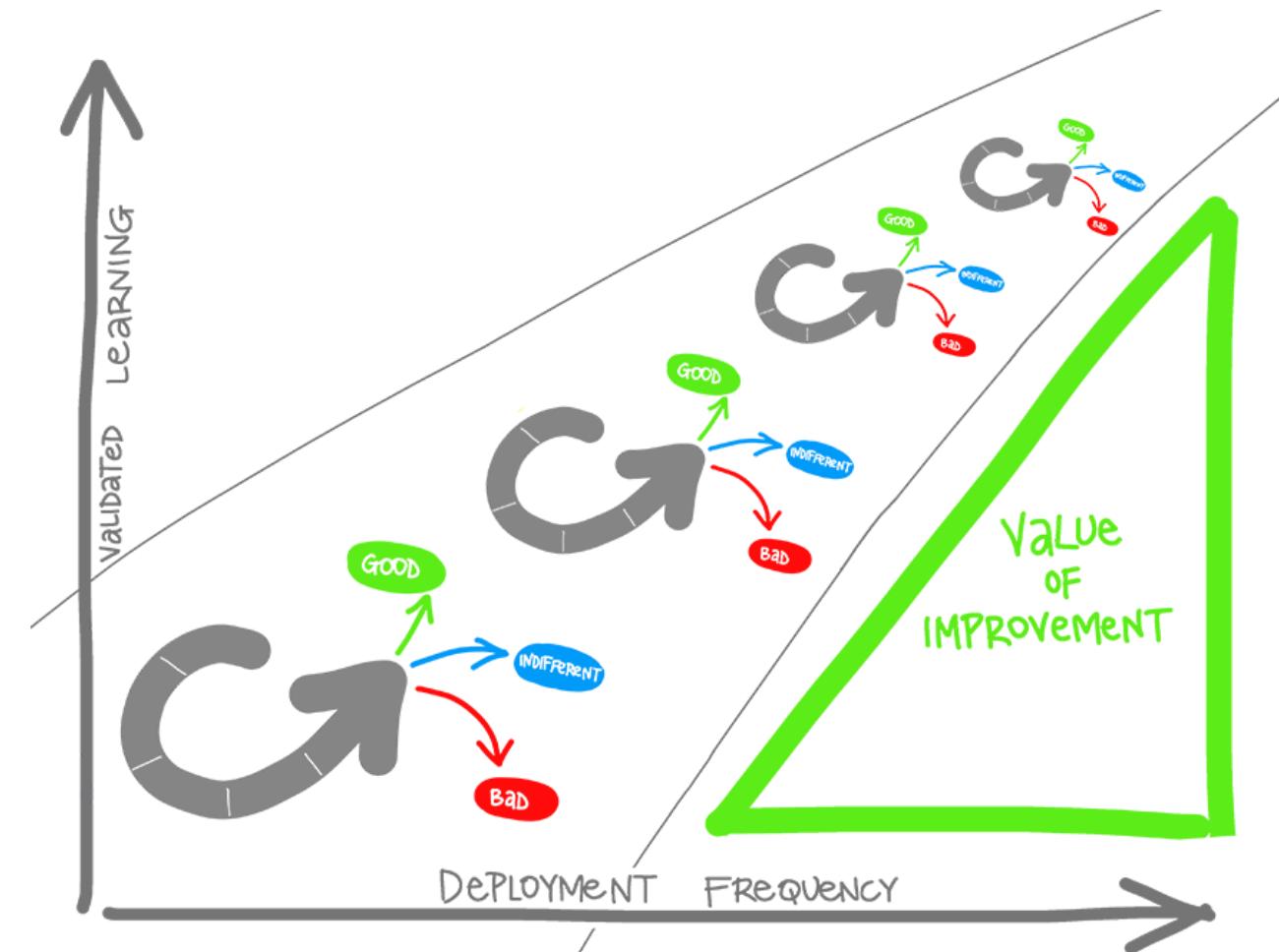
# Shorten your Cycle Time

- Adopt **DevOps** practices to continuously improve
- Work in smaller batches
- Use **automation**
- Improve your **telemetry** (gathering data from the application and environment)



# Optimize Validated Learning

- Frequent deployments lead to more experiments.
- Use experiments to understand good, bad, or indifferent results



# Testimonial #2: Why DevOps



## Release Pipeline

**VSO PRE-PROD SU0**  
VSO 2015.S81

1 day ago



**VSO PROD SU6**  
VSO 2015.S81

12 hours ago



**VSO PROD SU3**  
VSO 2015.S81

4 hours ago

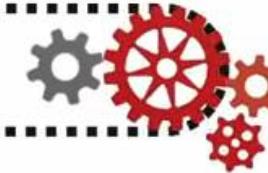


**VSO PROD SU4**  
VSO 2015.S81

Deploying since 2 mins



Visual Studio Online



# Case Study

How the Microsoft Azure DevOps Team, *formerly Visual Studio Team*, made their transformation from Waterfall to DevOps

# Azure DevOps



## Azure Boards

Deliver value to your users faster using proven agile tools to plan, track, and discuss work across your teams.



## Azure Pipelines

Build, test, and deploy with CI/CD that works with any language, platform, and cloud. Connect to GitHub or any other Git provider and deploy continuously.



## Azure Repos

Get unlimited, cloud-hosted private Git repos and collaborate to build better code with pull requests and advanced file management.



## Azure Test Plans

Test and ship with confidence using manual and exploratory testing tools.



## Azure Artifacts

Create, host, and share packages with your team, and add artifacts to your CI/CD pipelines with a single click.

**Extensions Marketplace**  
Access extensions from Slack to SonarCloud to 1,000 other apps and services—built by the community.



<https://azure.microsoft.com>

# The Azure DevOps Team

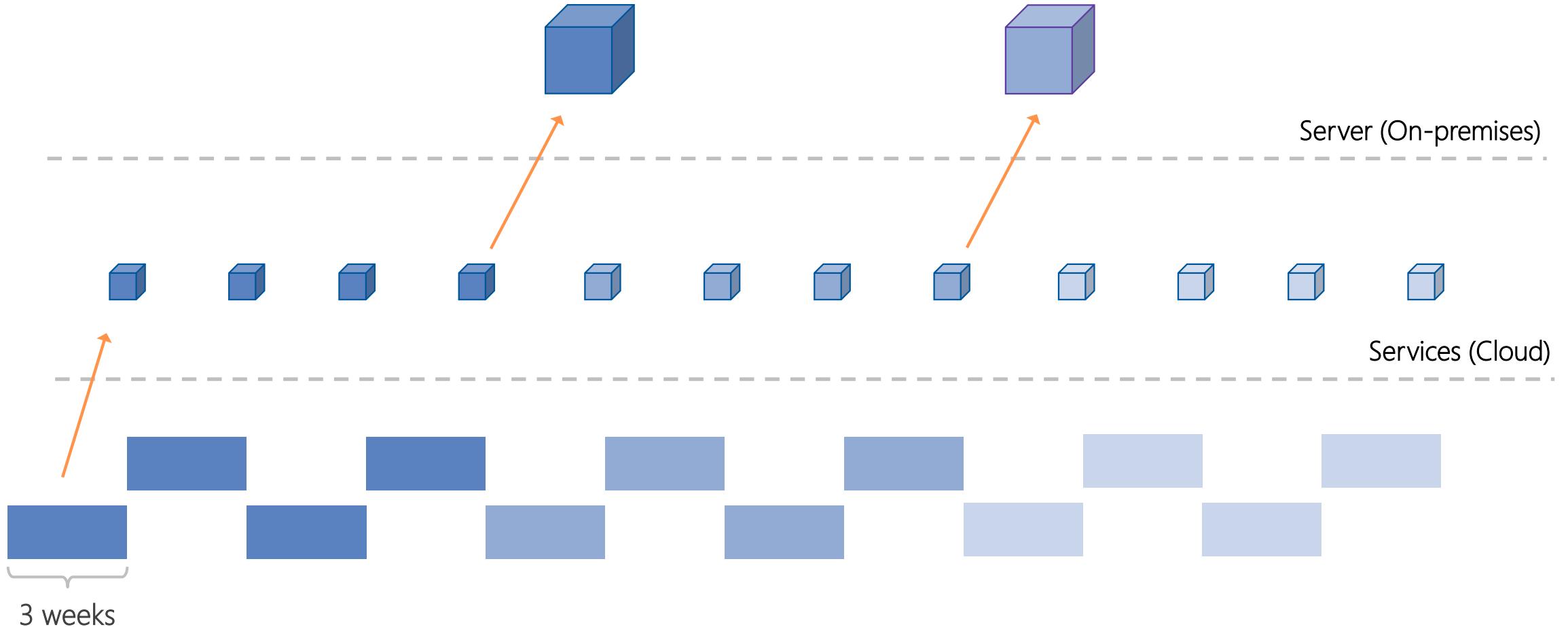
Redmond, Washington

- About 800 people
- 40 Feature Teams
- 3 Locations

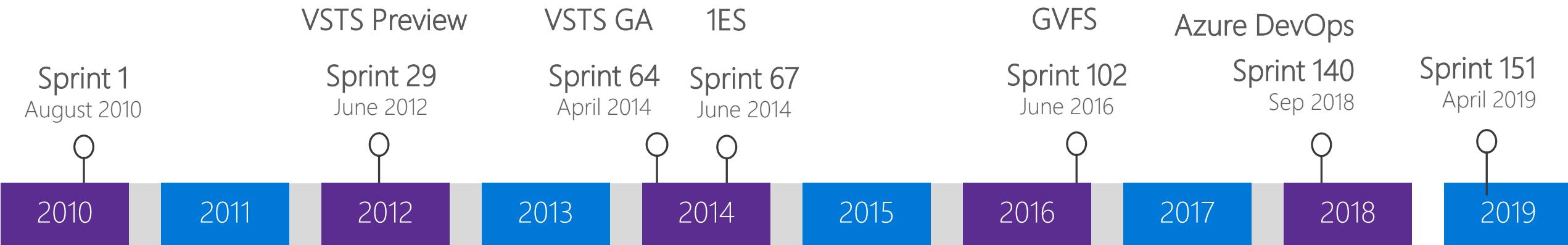


[This Photo](#) by Unknown Author is licensed under [CC BY-NC](#)

# How do we work?



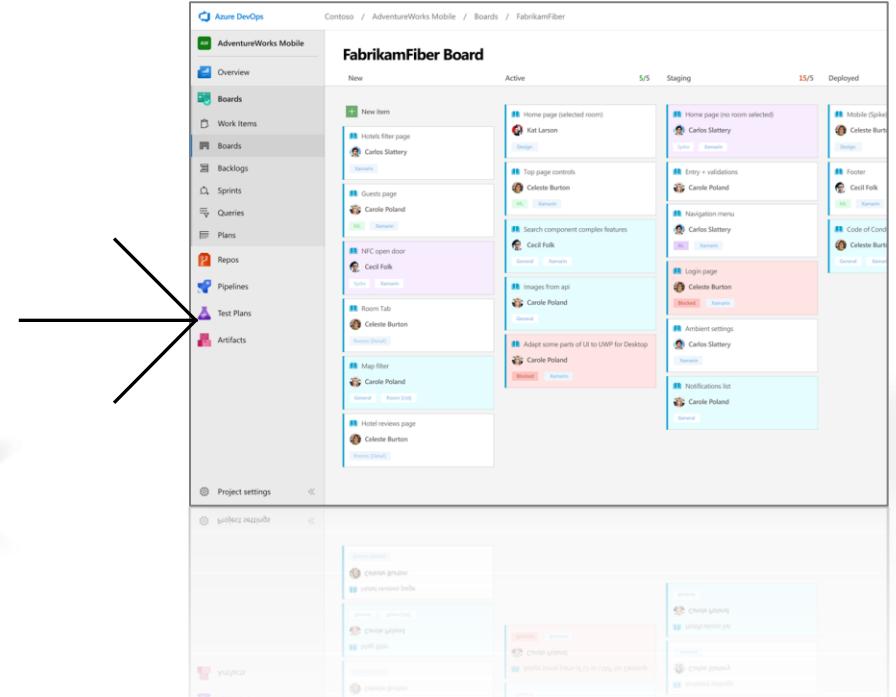
# Journey to DevOps



VSTS – Visual Studio Team Services  
GA – General Availability

1ES – One Engineering System  
(mandate to empower every engineer in the company by standardizing on the best available tools)

GVFS – Git Virtual File System  
(virtualizes the file system beneath your repo and makes it appear as though all the files in your repo are present, but only downloads a file the first time it is opened)



What did it look like before?

# Before

The OLD way



2 years

1 year

# Before

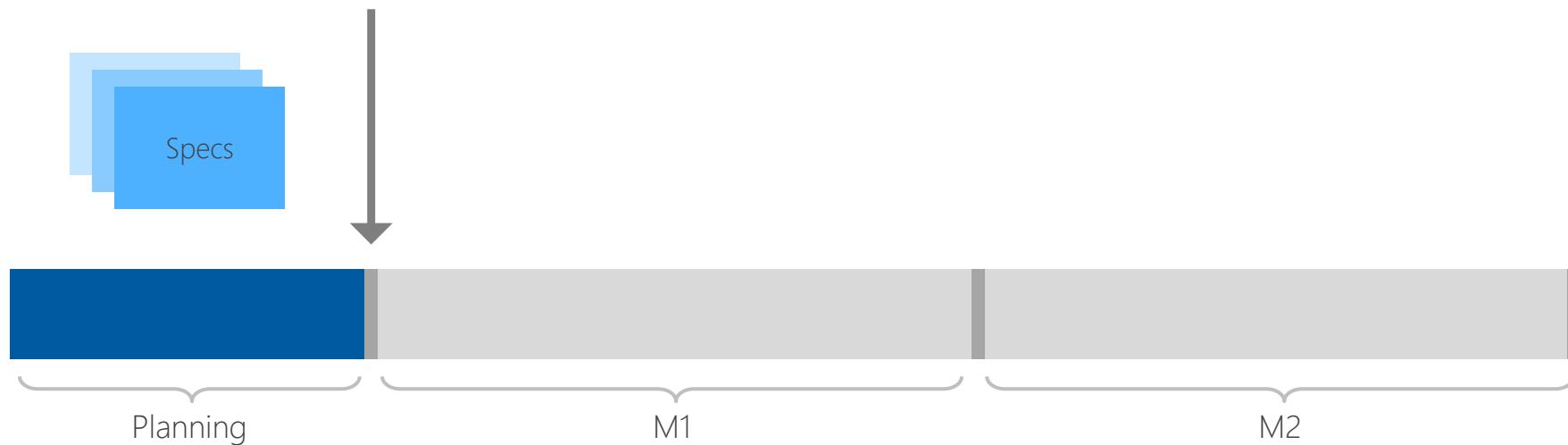
The OLD way



# Before

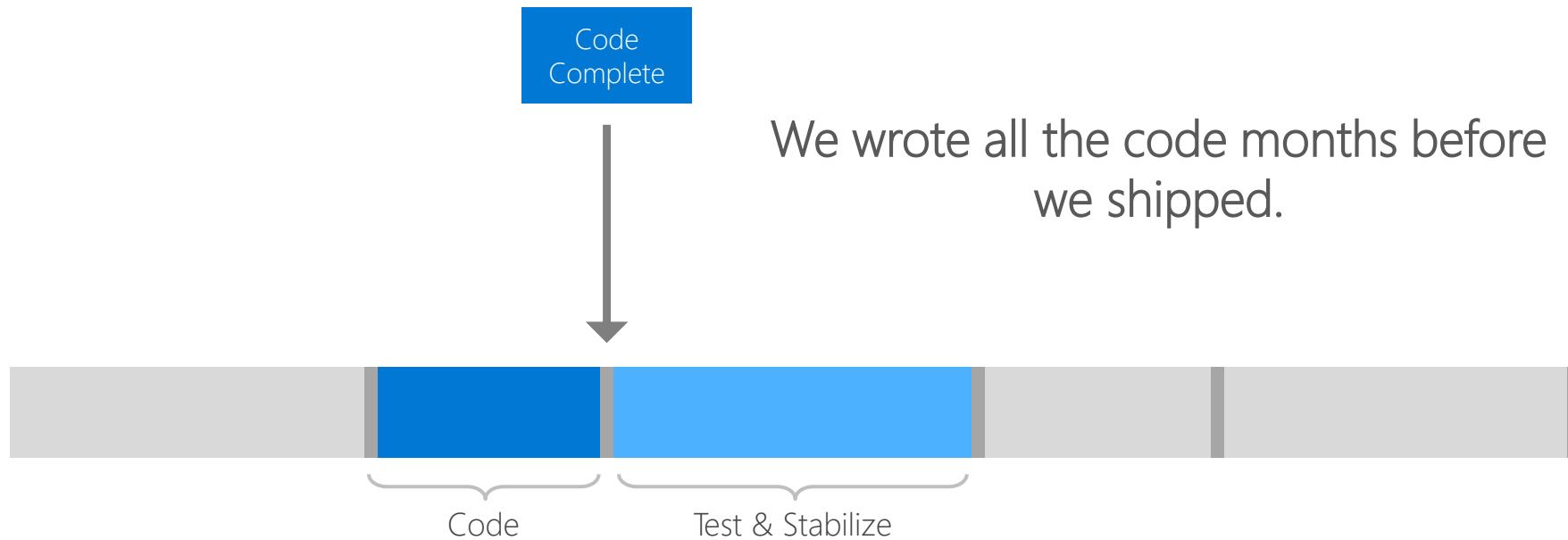
The OLD way

We knew exactly what to build...  
and we knew it was right!



# Before

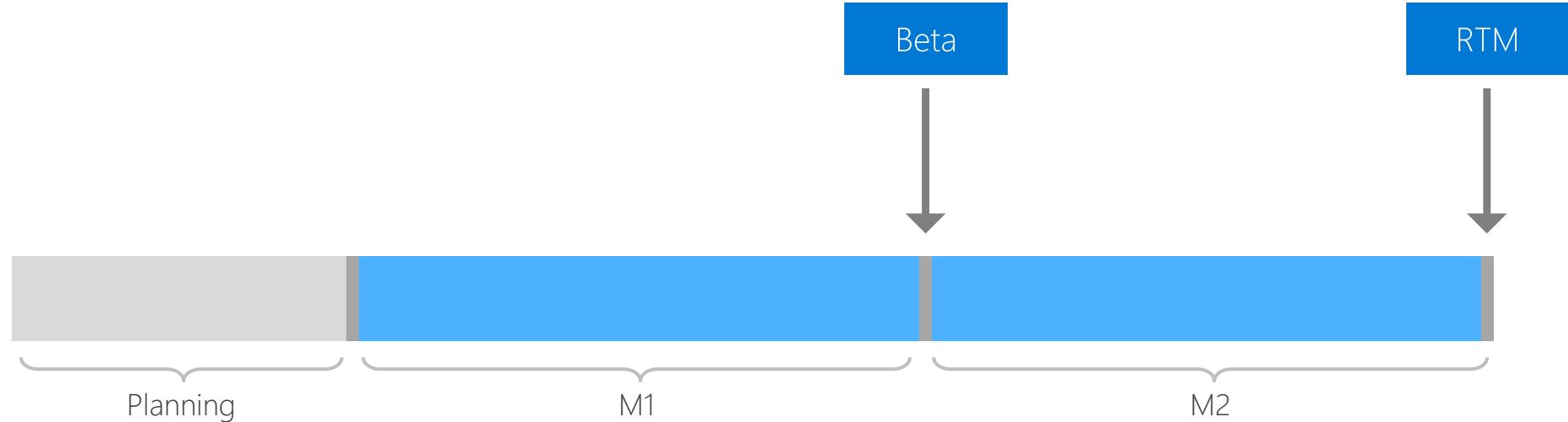
The OLD way



# Before

The OLD way

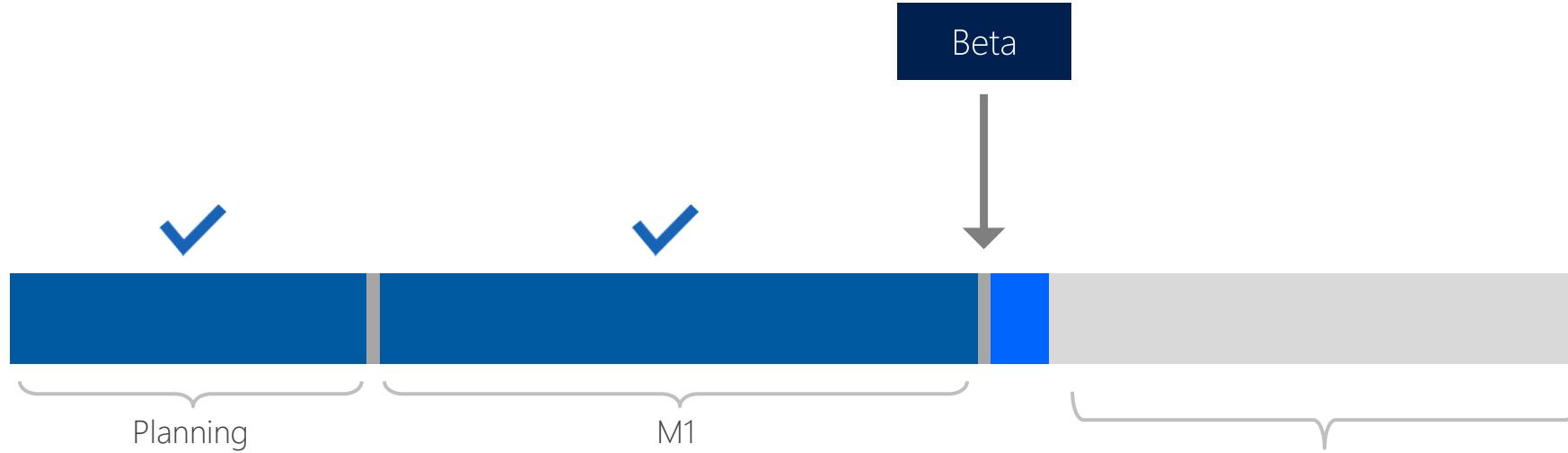
We had a perfect schedule and knew exactly when it would be ready!



# Before

The OLD way

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

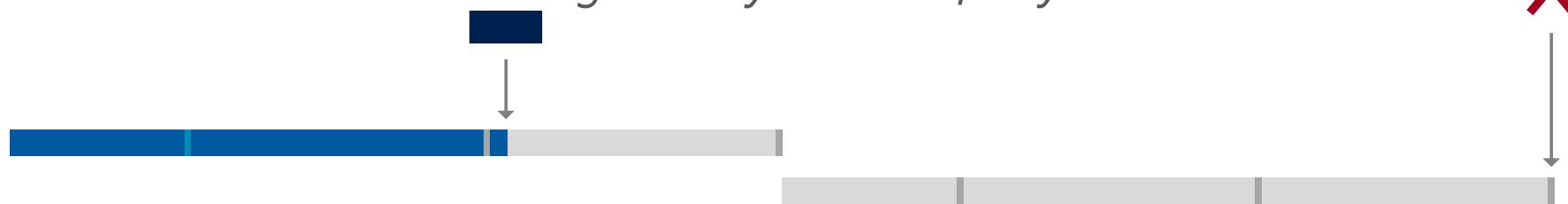


... but we're booked solid already.

# Before

The OLD way

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



Question: How did it work?

Very well in the era in which it was born. However...

“Firms today experience a much higher velocity of business change. Market opportunities appear or dissolve in months or weeks instead of years.”

Diego Lo Giudice and Dave West, Forrester

February 2011

Transforming Application Delivery

What changed?

# Before and After

4 to 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

3-week sprints

Vertical teams

Team rooms

Continual Planning & Learning

PM & Engineering

Continual customer engagement

Everyone in main

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

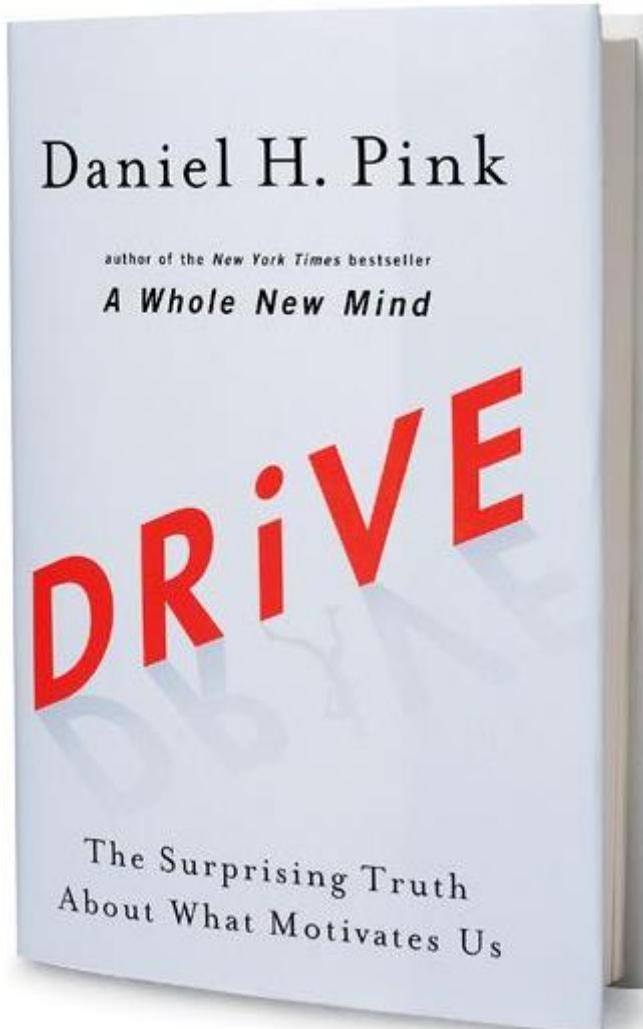
# Instrumental Changes

- Changed our culture
  - Autonomy, mastery, and purpose
- Changed our approach to teams
  - Program Management is the WHAT and WHY; Engineering is HOW with Quality and Security
- Changed how we plan & learn
- Created new ways to stay healthy
  - Introduced bug cap

“Culture eats strategy for breakfast.”

Peter Drucker

# Motivation

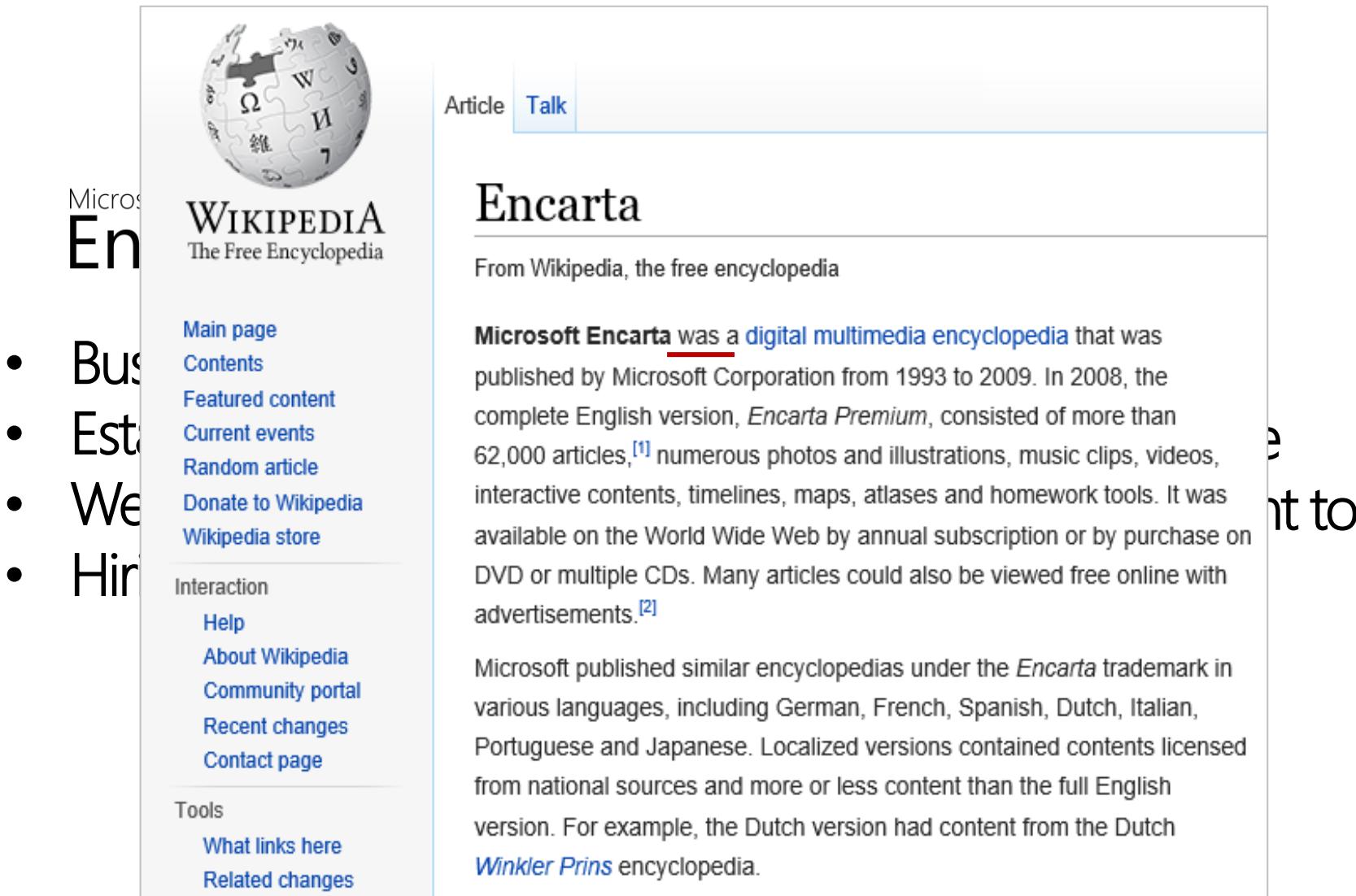


"Most people believe that the best way to motivate is with rewards like money—the carrot-and-stick approach. That's a mistake, says Daniel H. Pink (author of *To Sell Is Human: The Surprising Truth About Motivating Others*). In this provocative and persuasive new book, he asserts that the secret to high performance and satisfaction—at work, at school, and at home—is the deeply human need to direct our own lives, to learn and create new things, and to do better by ourselves and our world." (Pink, 2020)



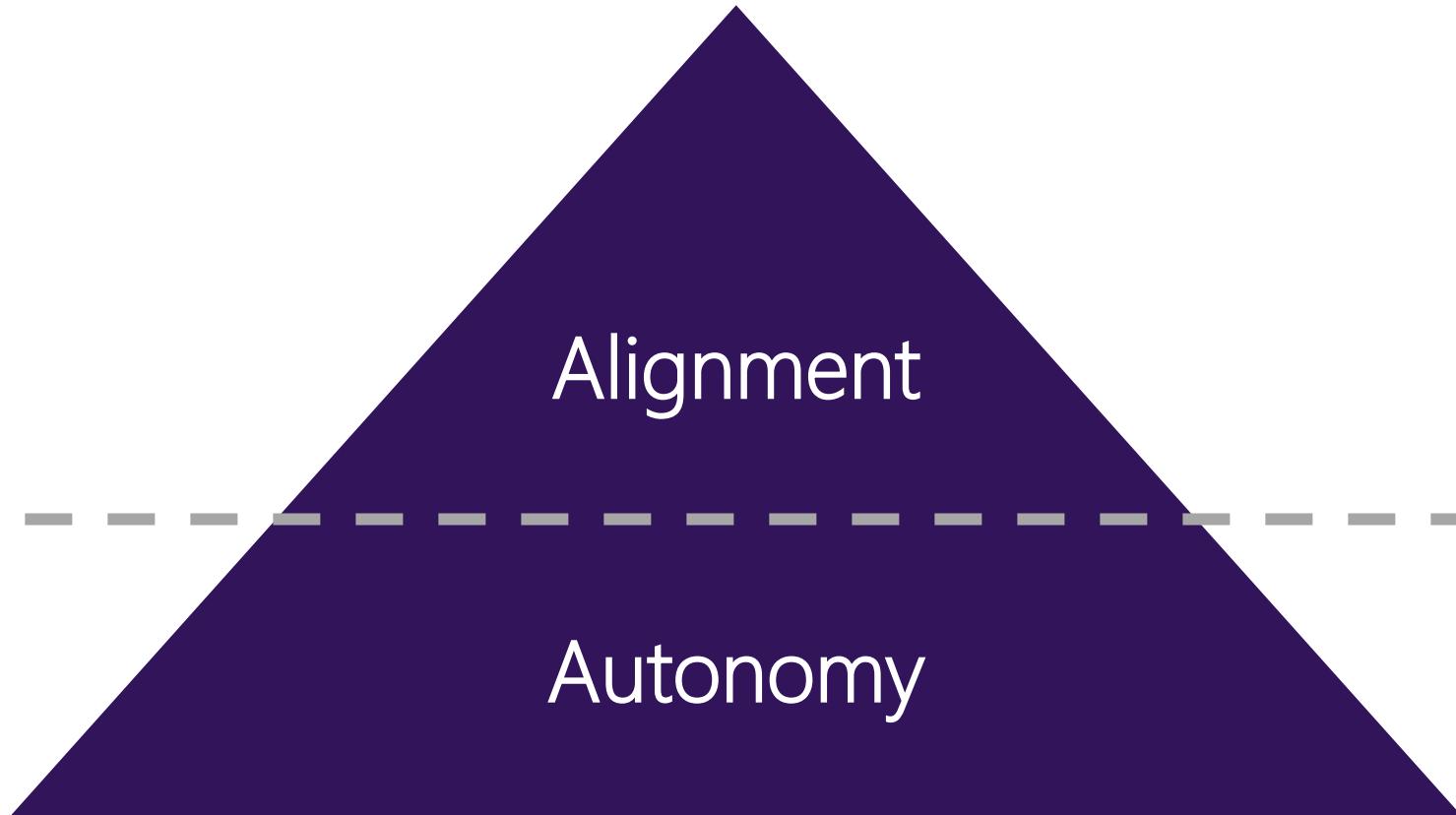
*Let's try to give our teams three things: Autonomy, Mastery, and Purpose.*

# Autonomy, Mastery, and Purpose

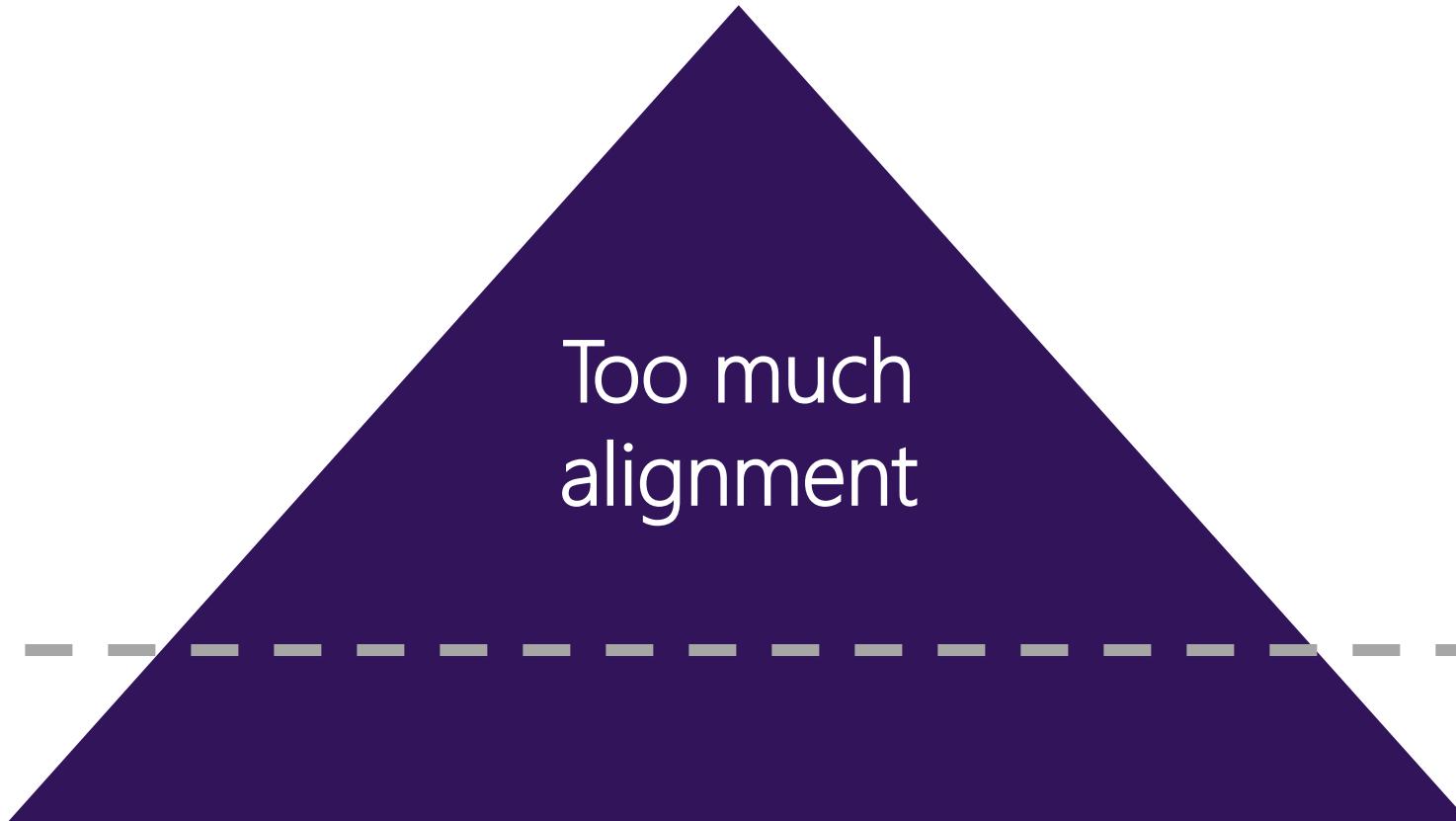


The image is a screenshot of a Wikipedia article page. At the top left is the Wikipedia logo (a globe made of puzzle pieces) and the word "WIKIPEDIA" with the tagline "The Free Encyclopedia". To the right are two tabs: "Article" (which is selected) and "Talk". The main title of the article is "Encarta". Below the title, it says "From Wikipedia, the free encyclopedia". The article text begins with: "Microsoft **Encarta** was a [digital multimedia encyclopedia](#) that was published by Microsoft Corporation from 1993 to 2009. In 2008, the complete English version, *Encarta Premium*, consisted of more than 62,000 articles, [1] numerous photos and illustrations, music clips, videos, interactive contents, timelines, maps, atlases and homework tools. It was available on the World Wide Web by annual subscription or by purchase on DVD or multiple CDs. Many articles could also be viewed free online with advertisements. [2]" Below this, there is a paragraph about Microsoft publishing similar encyclopedias in other languages. On the left side of the page, there is a sidebar with various links: "Main page", "Contents", "Featured content", "Current events", "Random article", "Donate to Wikipedia", "Wikipedia store", "Interaction", "Help", "About Wikipedia", "Community portal", "Recent changes", "Contact page", "Tools", "What links here", and "Related changes".

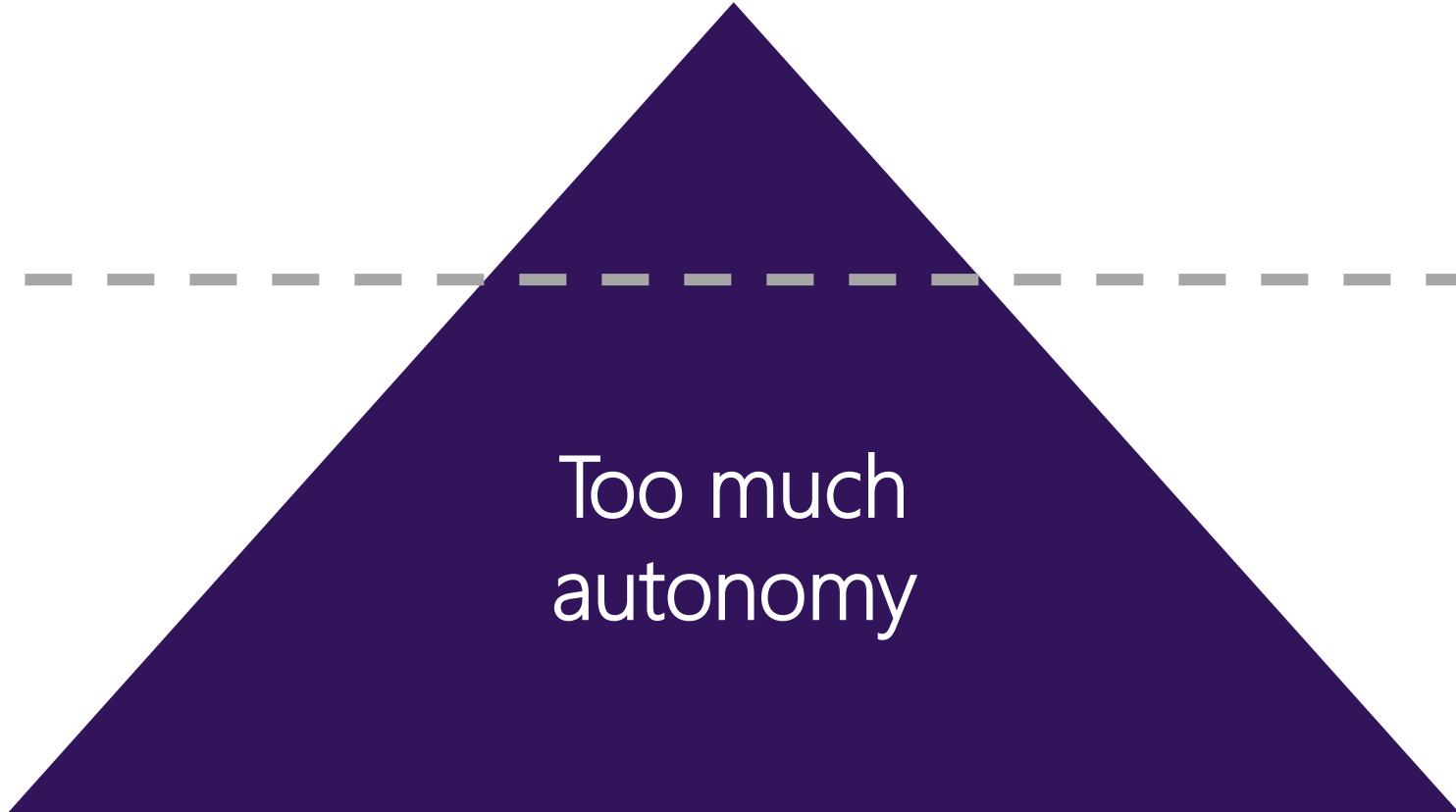
# Aligned Autonomy



# Aligned Autonomy



# Aligned Autonomy

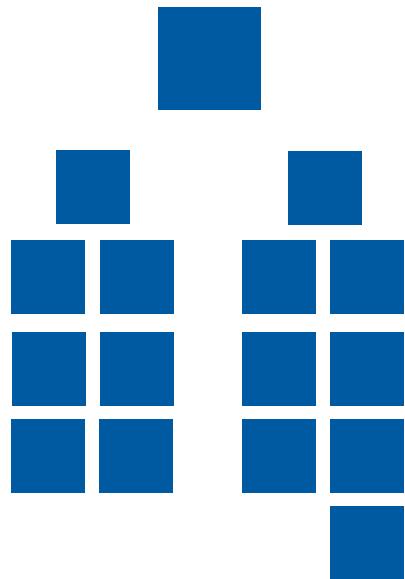


Let's look at some of the notable changes...

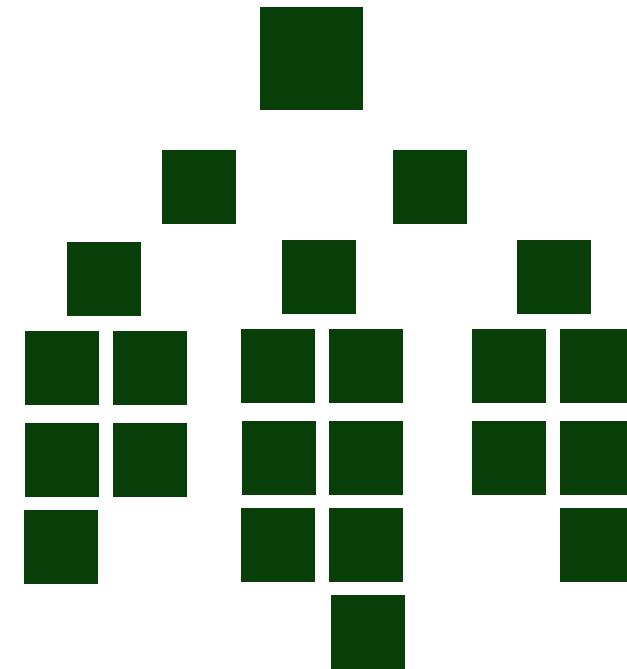
# Roles

# The OLD way

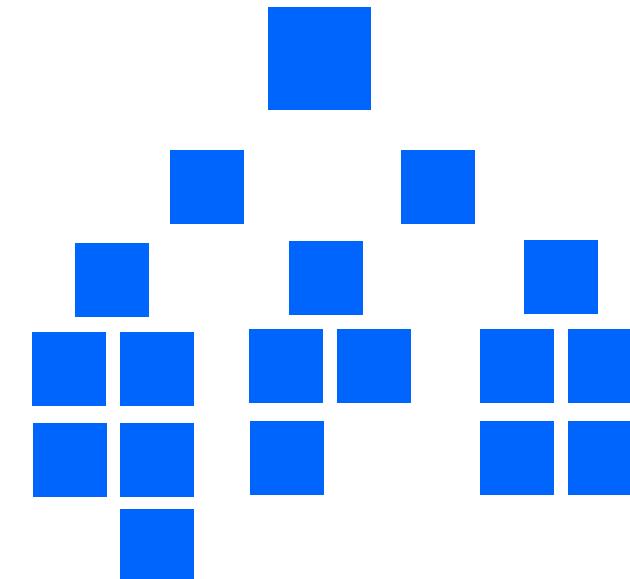
# Program Management



Dev

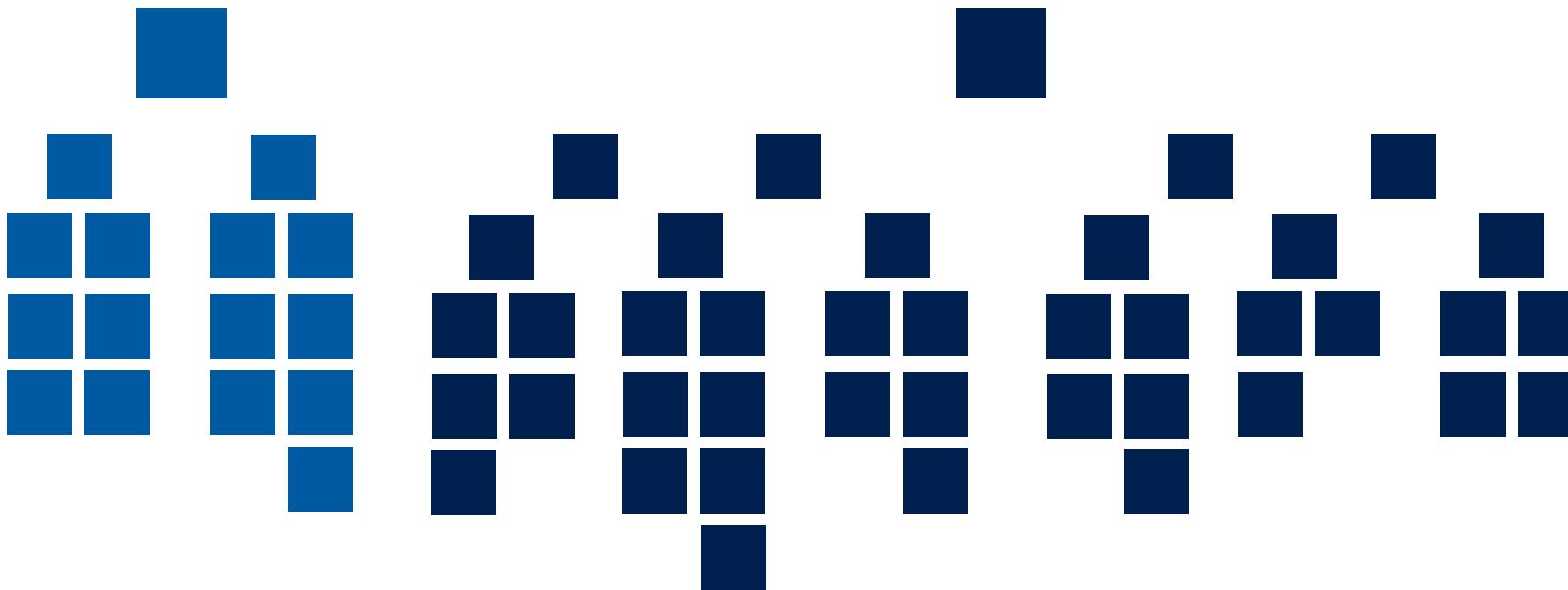


Test



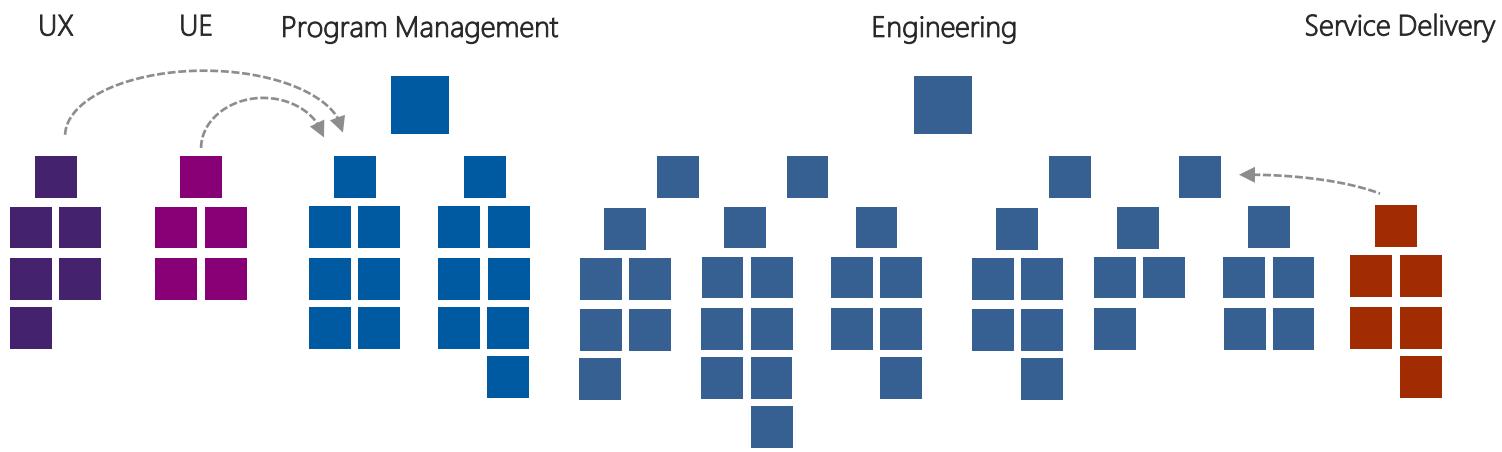
# Roles

# Program Management



# Engineering

# Roles

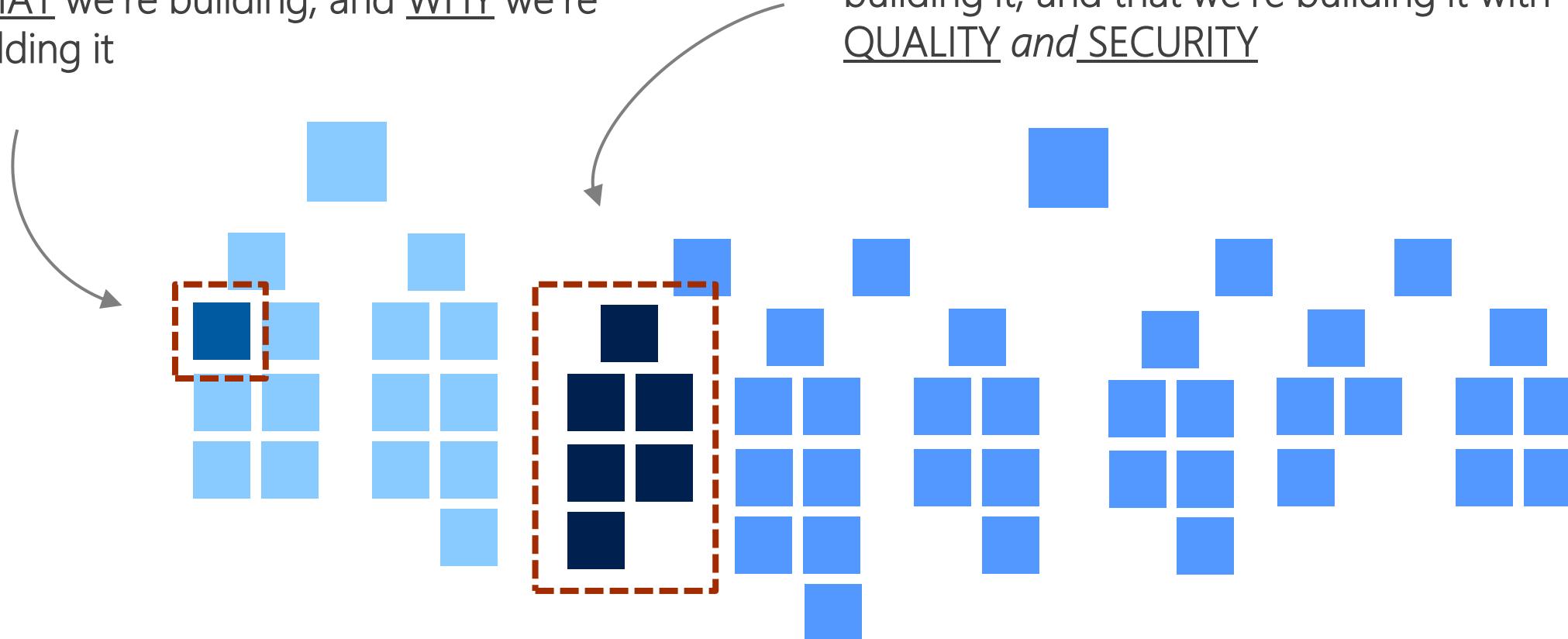


Service Delivery is integrated directly into our organization.

# Teams

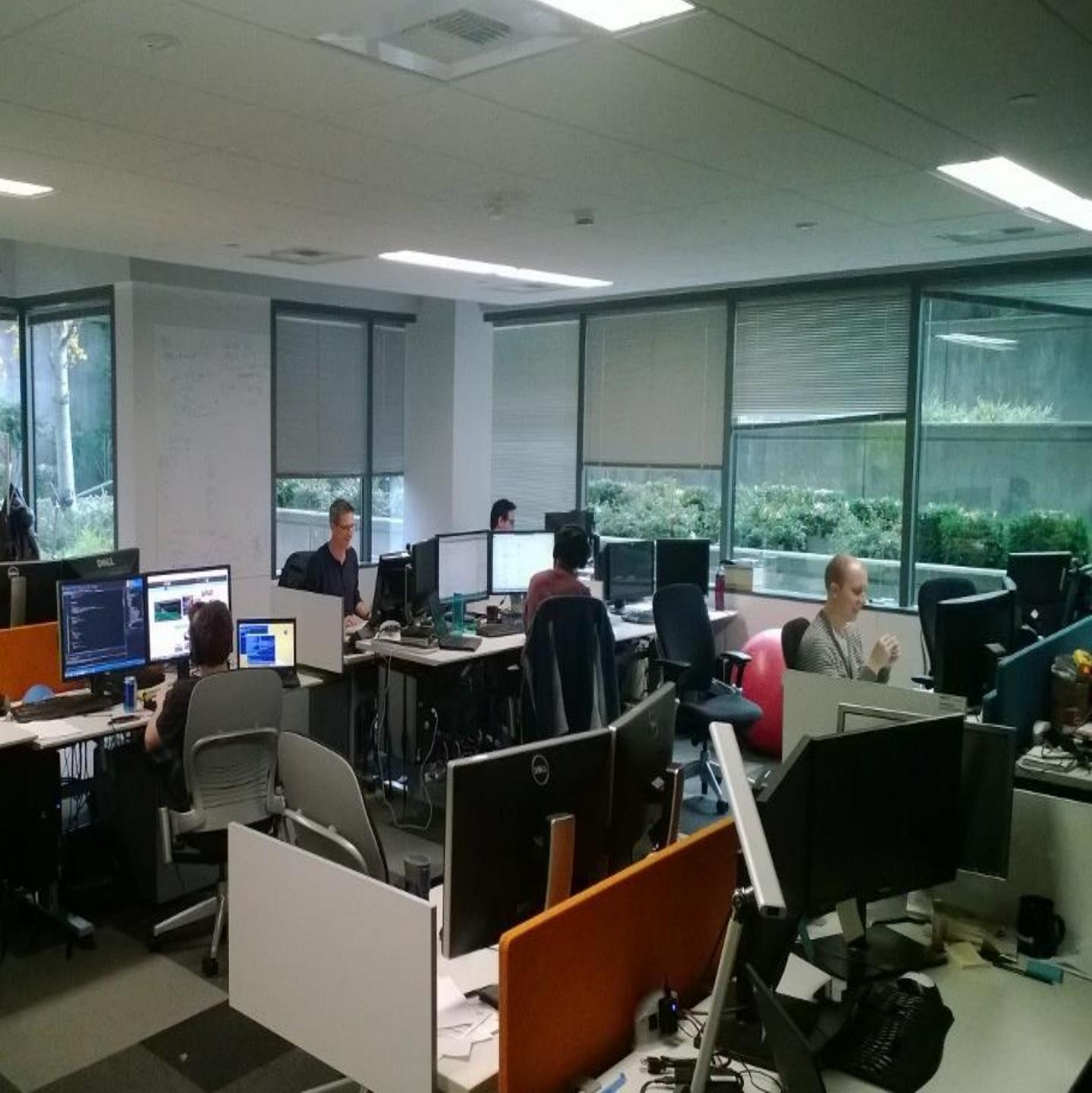
Program Management is responsible for:  
WHAT we're building, and WHY we're  
building it

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

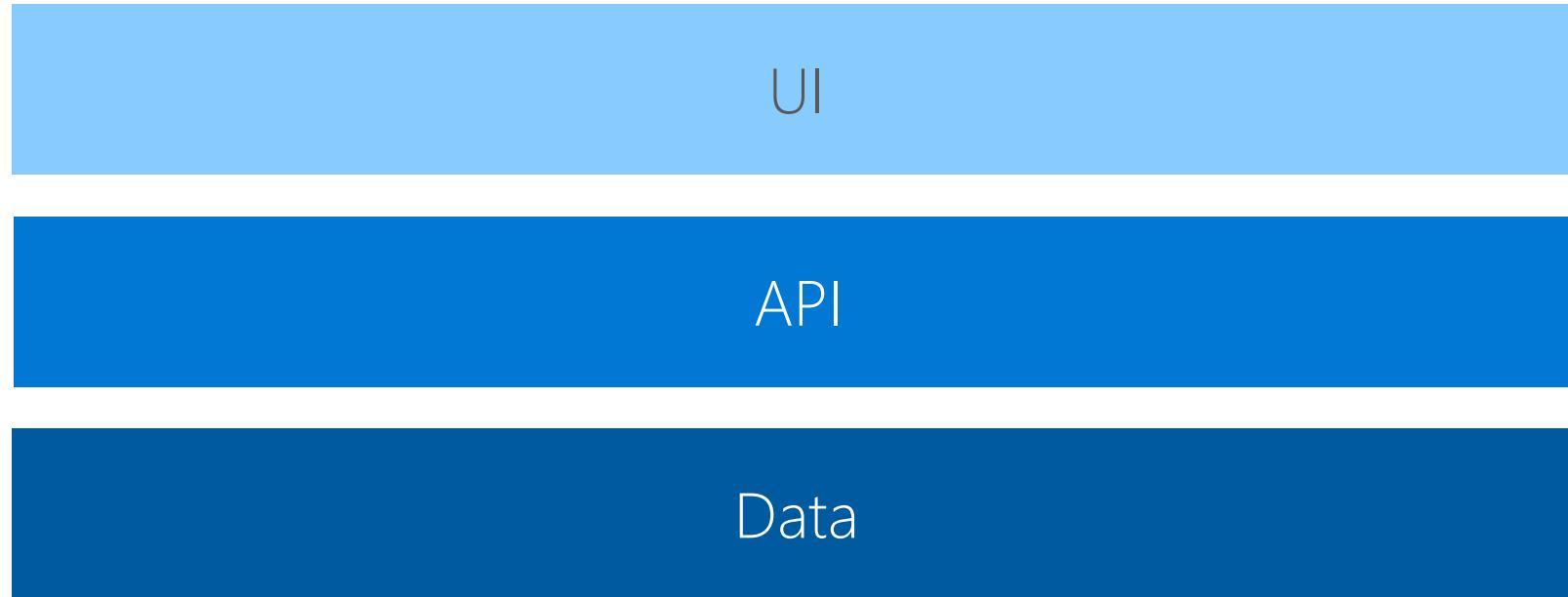


# Team Rooms

- Cross discipline
- 10-12 people
- Self managing
- Clear charter and goals
- Intact for 12-18 months
- Physical team rooms
- Own features in production
- Own deployment of features

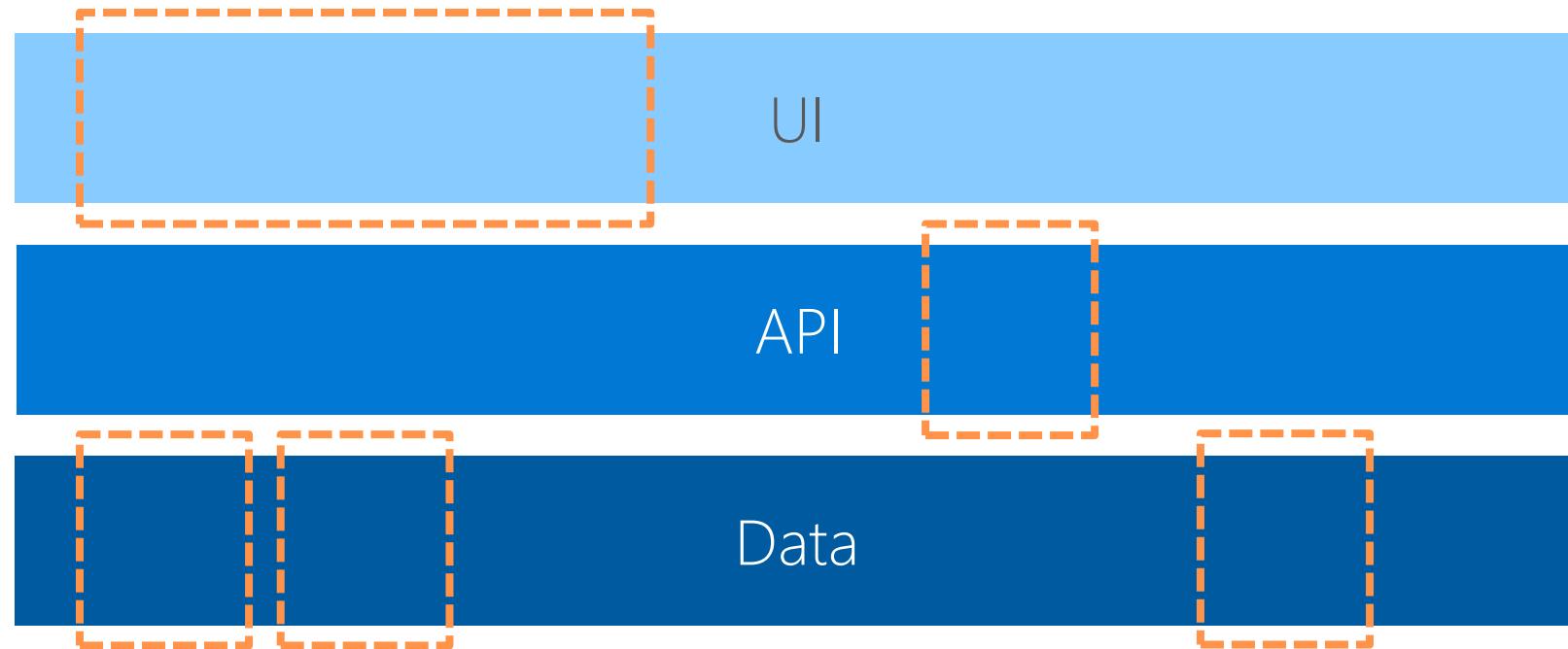


# Team Composition

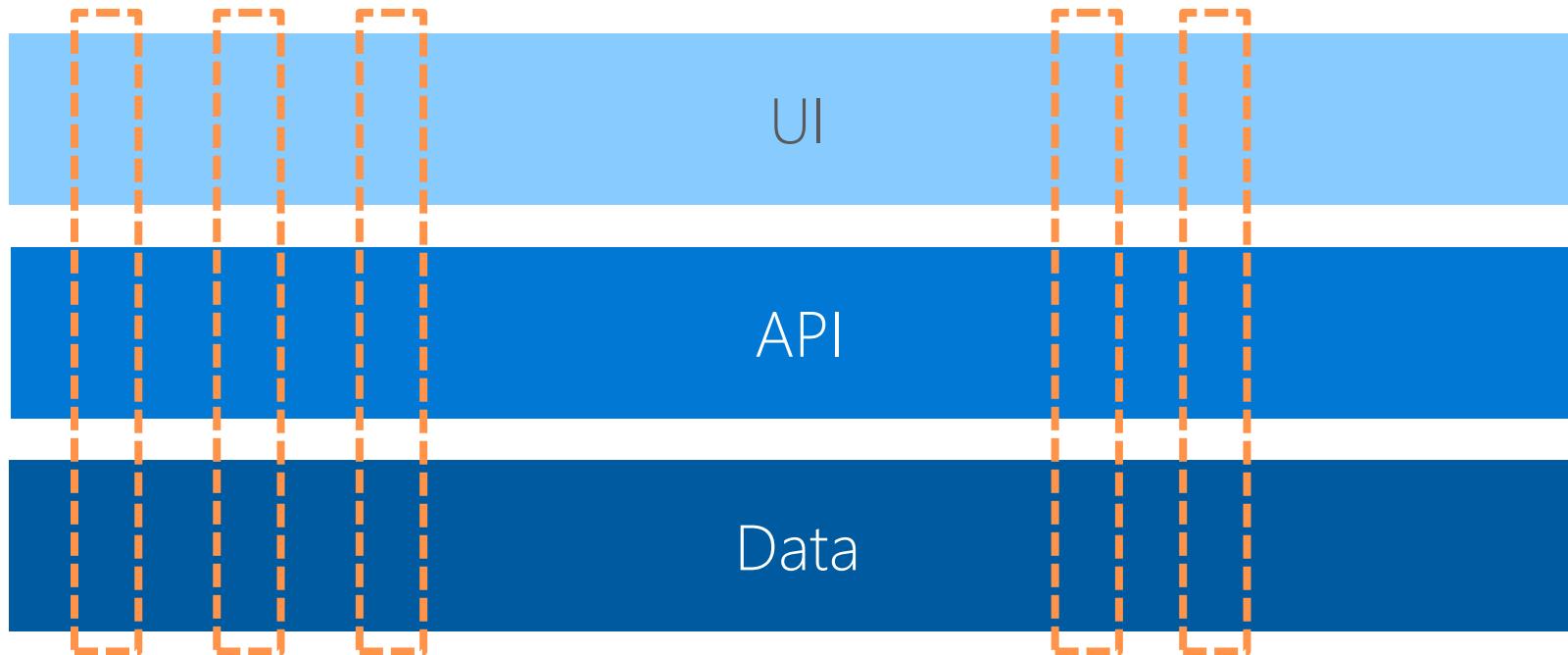


# Horizontal Composition

The OLD way



# Vertical Composition

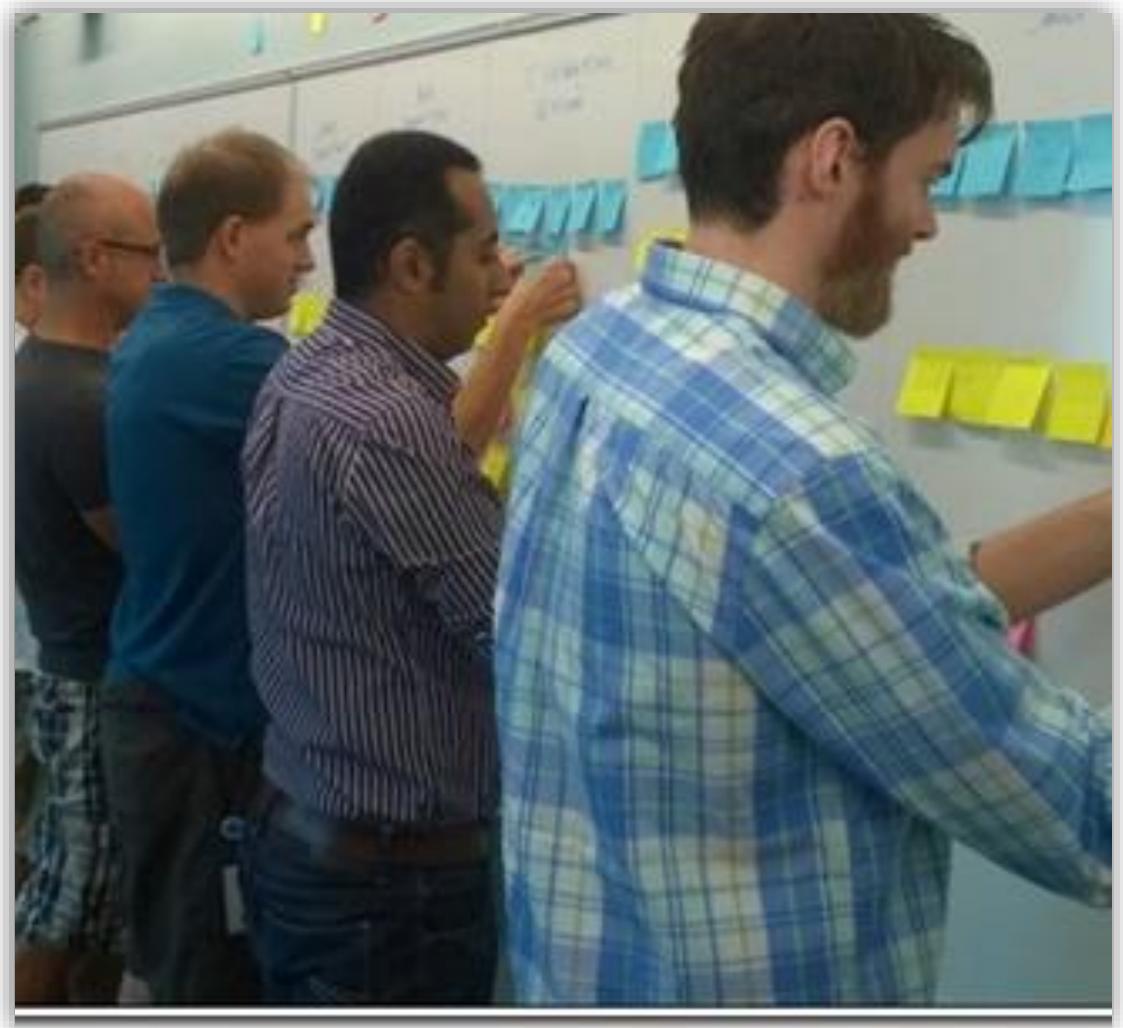


# Self-Organizing Teams

- We have chosen to re-think the charter and make-up of our teams at strategic checkpoints. This occurs about every 18 months (about 1 and a half years).
- Yellow Sticky Exercise:
- Autonomy – Let the team choose what they want to work on
- Alignment – Ensure we have the right balance

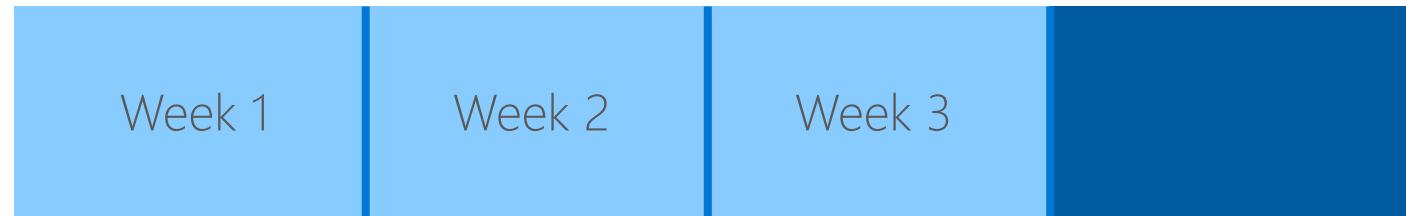
<http://aka.ms/selfformingteams>

Blog post from Brian Harry, Corporate Vice President, describing the process



How do teams stay connected?

# Three Week Sprints



Sprint 113



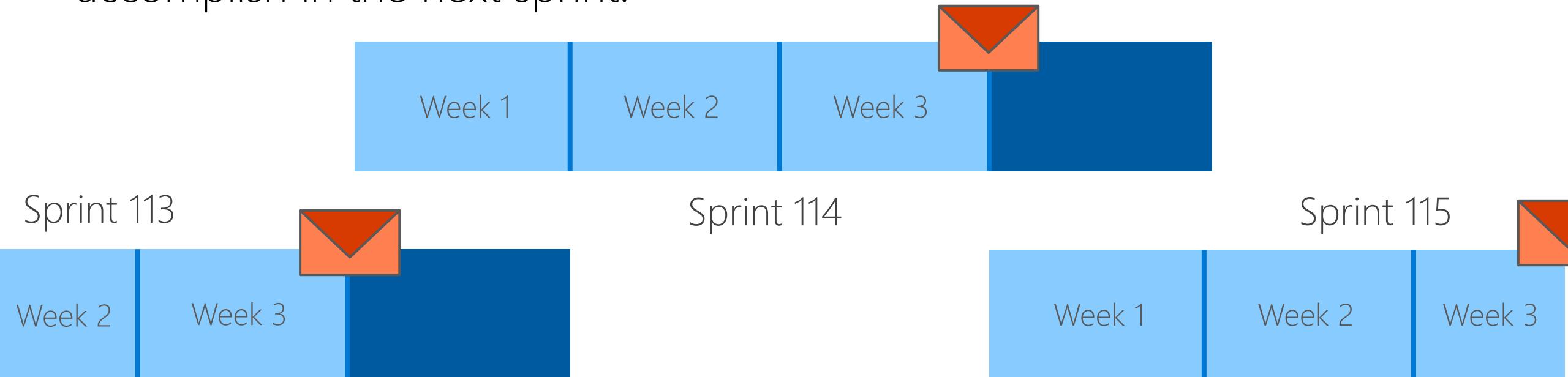
Sprint 114



Sprint 115

# Sprint Mail

At the end of a print, all teams send a “sprint email” that communicates what they have accomplished in the sprint, and what they are planning to accomplish in the next sprint.

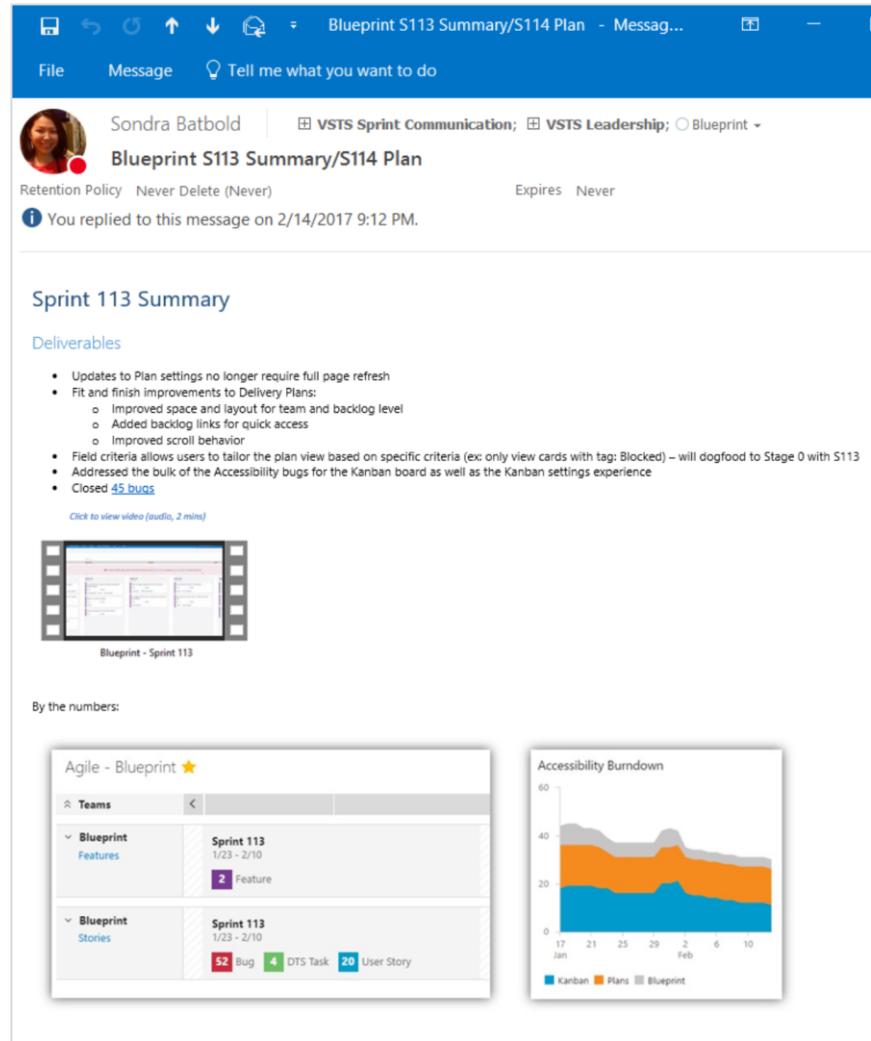


# Sprint Mail

✉ 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 - Message...

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 min)

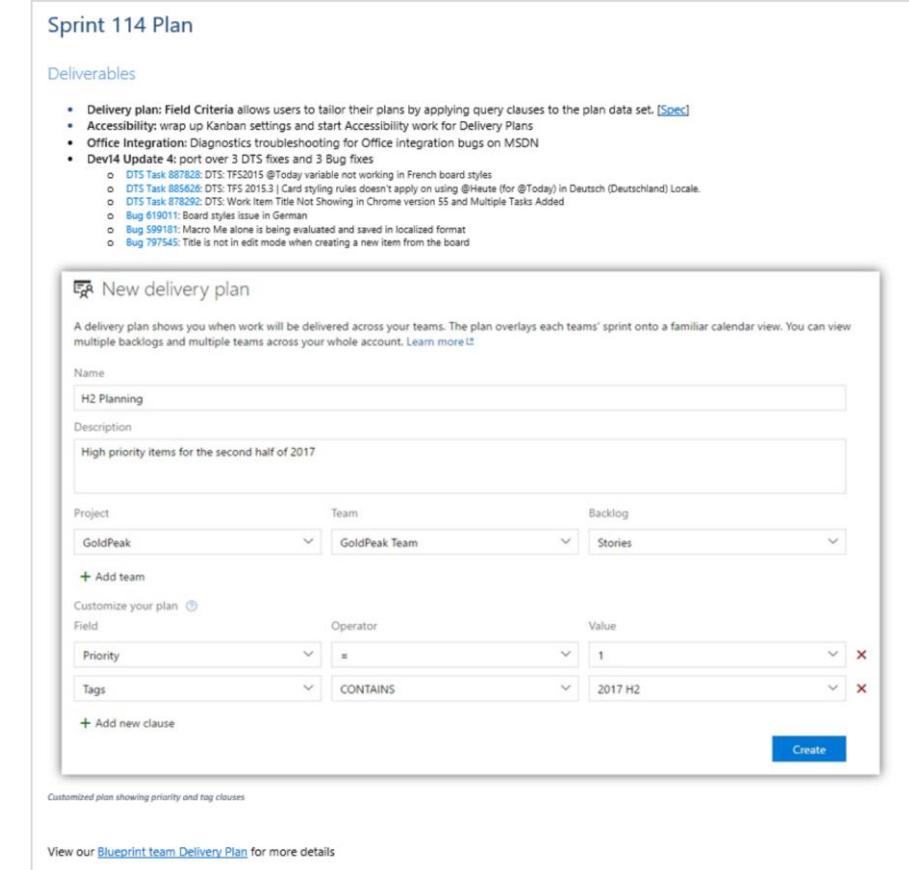
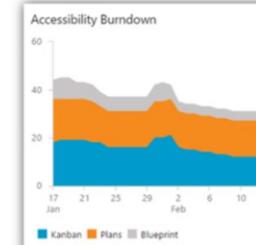
Blueprint - Sprint 113

#### By the numbers:

Agile - Blueprint ★

Teams	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



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 backlog and multiple teams across your whole account. Learn more ↗

Name: H2 Planning

Description: High priority items for the second half of 2017

Project	Team	Backlog
GoldPeak	GoldPeak Team	Stories

+ Add team

Customize your plan

Field	Operator	Value
Priority	=	1
Tags	CONTAINS	2017 H2

+ Add new clause

Customized plan showing priority and tag clauses

View our Blueprint team Delivery Plan for more details

## Benefits

- Minimize dependencies between teams and layers
- Teams learn how to work together
- Ensures cross-discipline and ownership
- Provides teams with maximum autonomy
- Progress is continually shared

“Plans are worthless, but planning is everything.”

Dwight Eisenhower

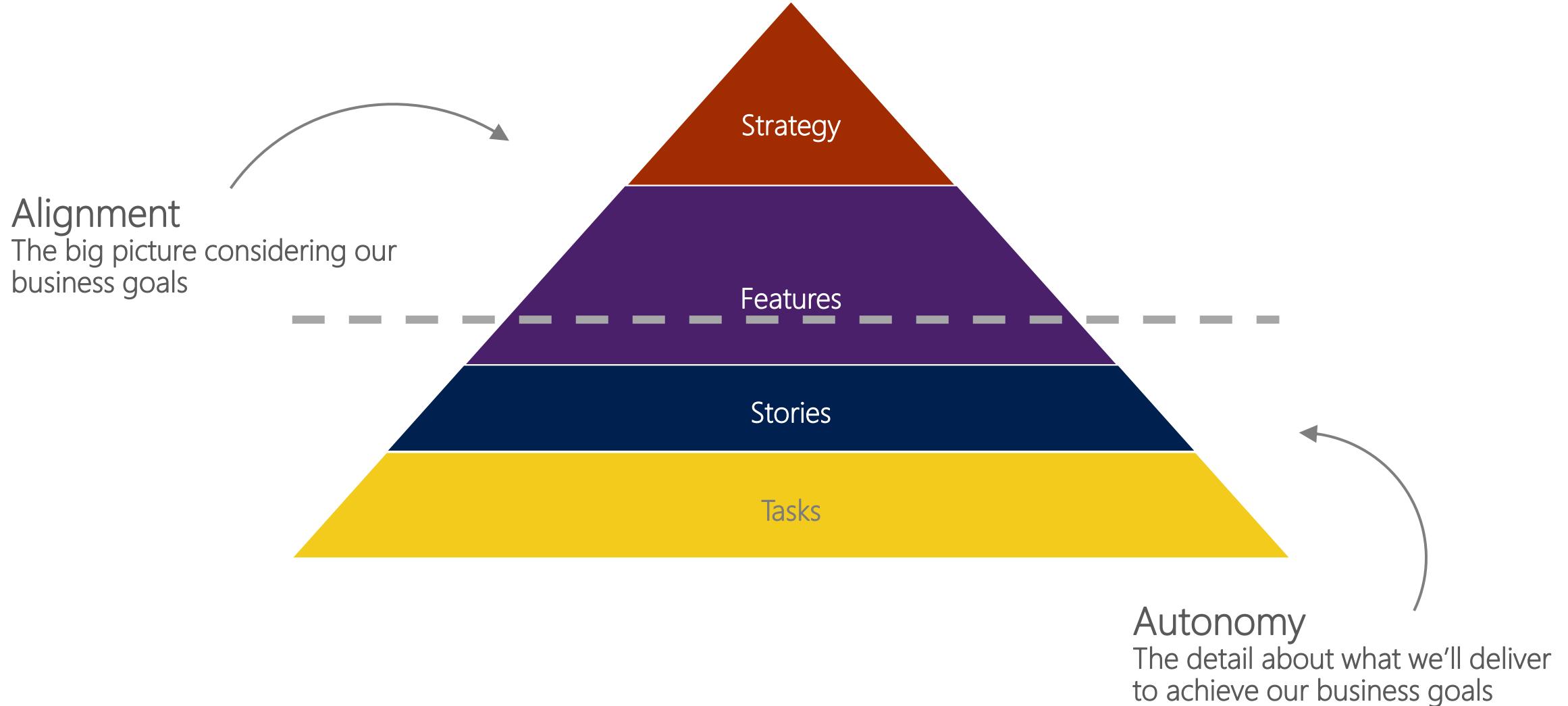
# Planning



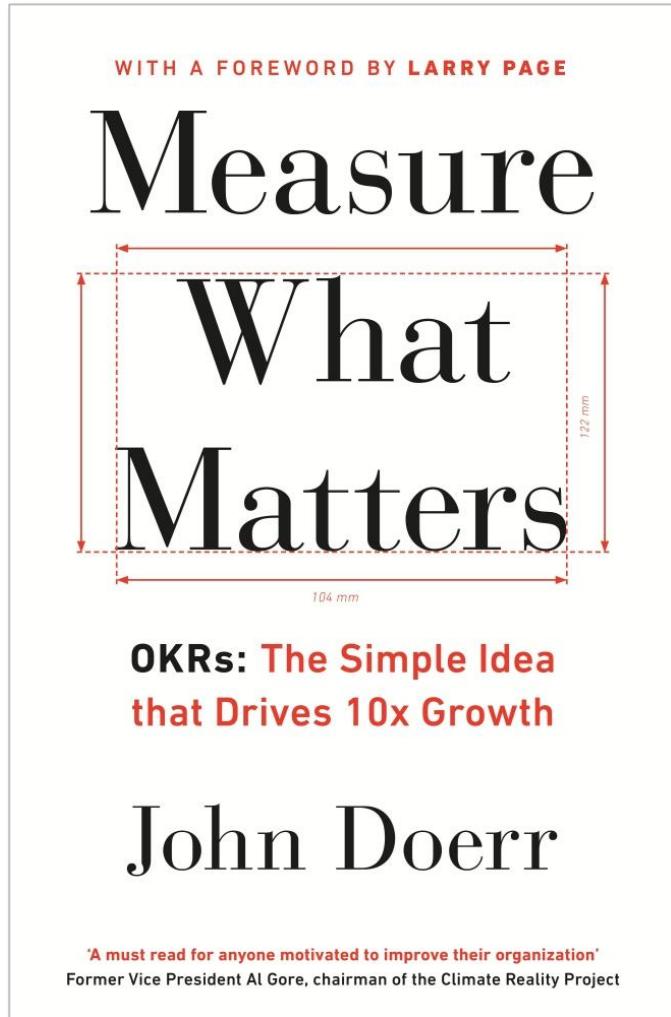
Leadership is responsible  
for the big picture

Teams are responsible for the detail

# Aligned Autonomy

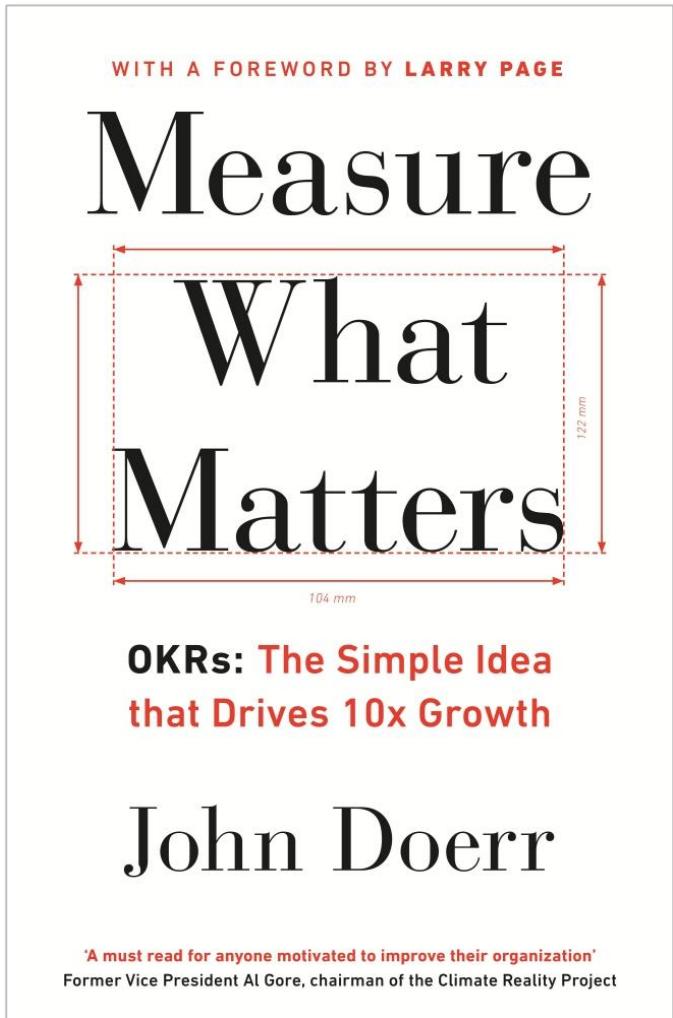


# Objectives and Key Results (OKRs)



- Objectives and Key Results (OKR) is a popular technique for setting and communicating goals and results in organizations.
- Its main goal is to connect company, team and personal objectives to measurable results, making people move together in right direction.

# Objectives and Key Results (OKRs)



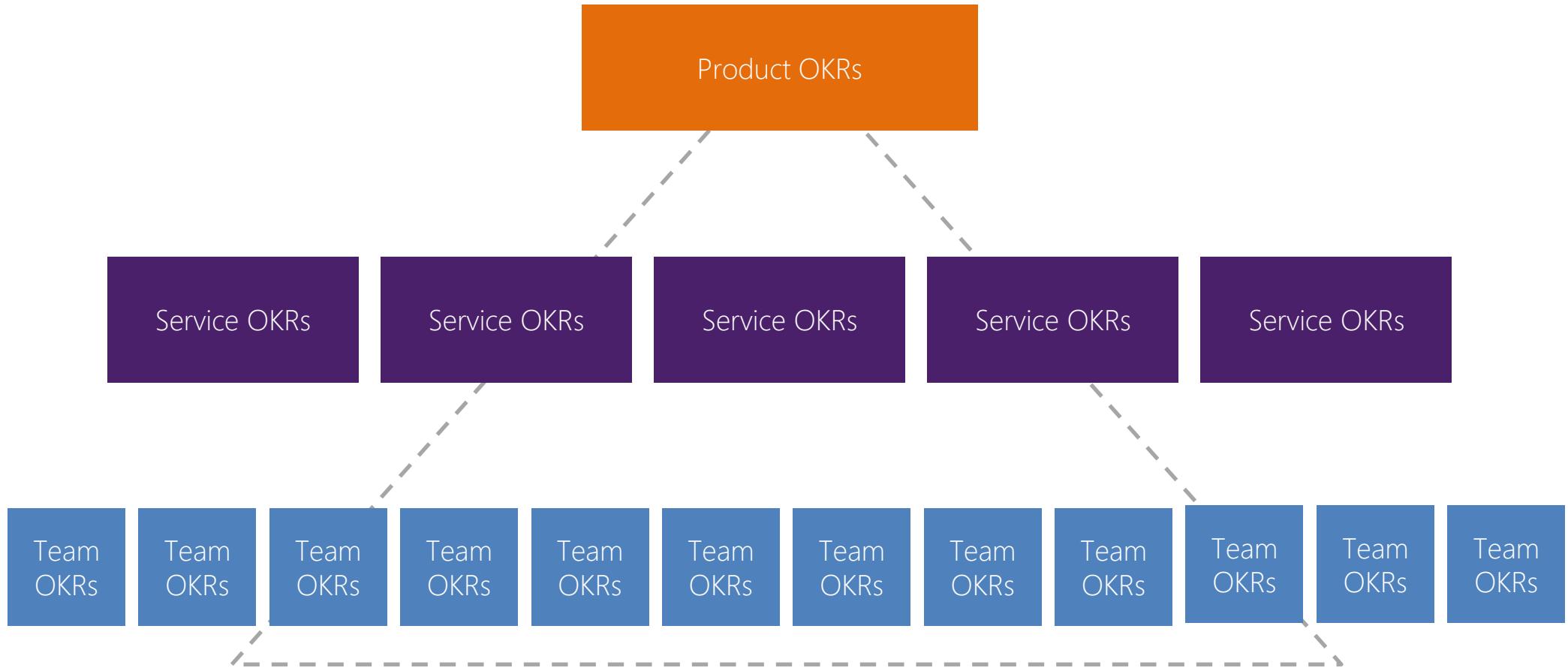
## Objective

Put an astronaut on the Moon by 1969

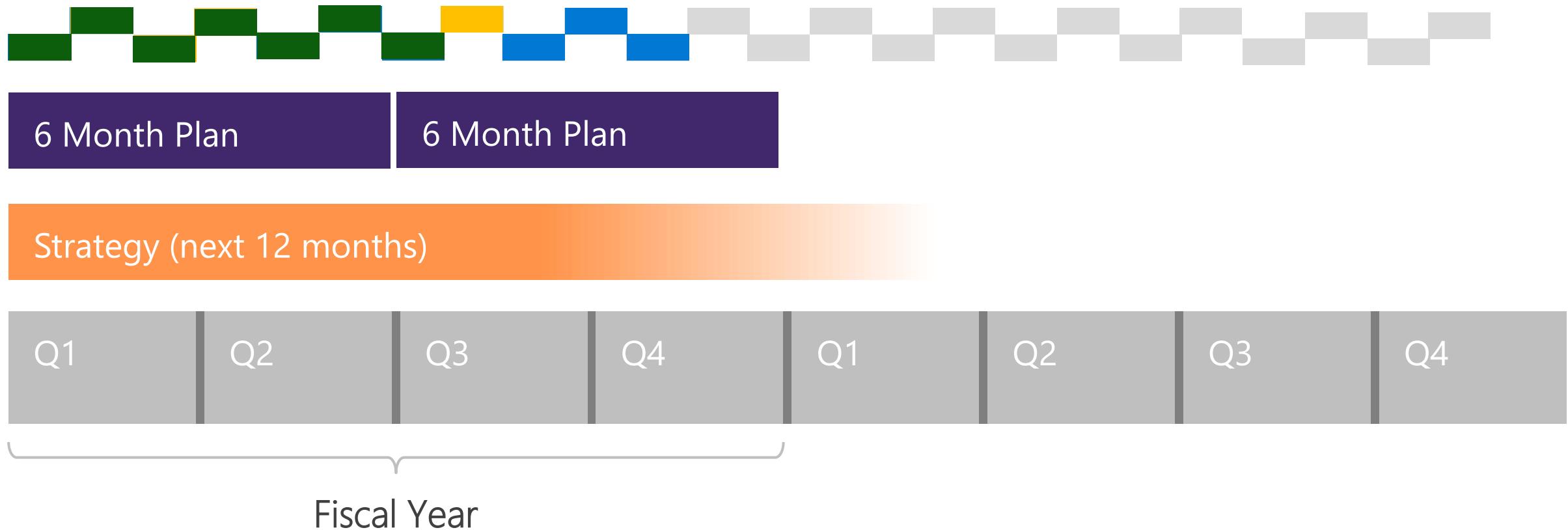
## Key Results

1. Build a spaceship under 40,000 pounds by 1965
2. Train astronauts for Moon landing by 1967
3. Successfully land the spaceship on the Moon
4. Safely bring the astronauts back to Earth

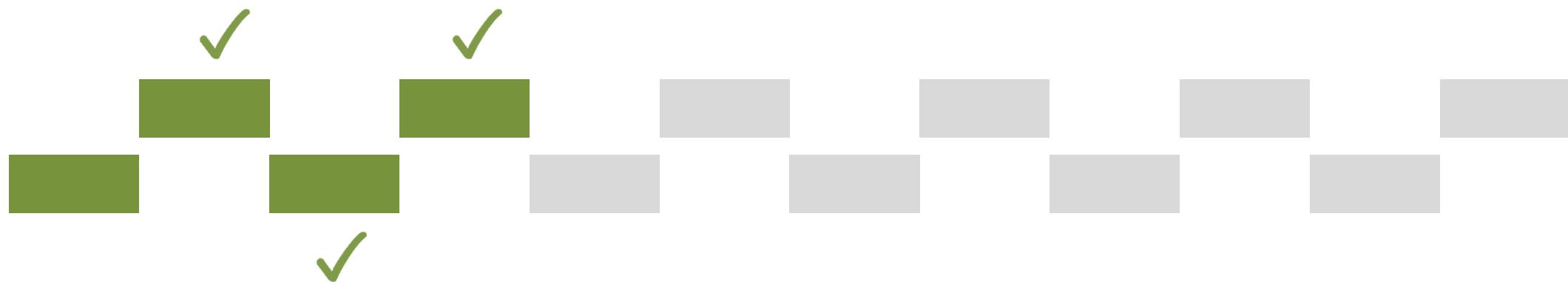
# Alignment



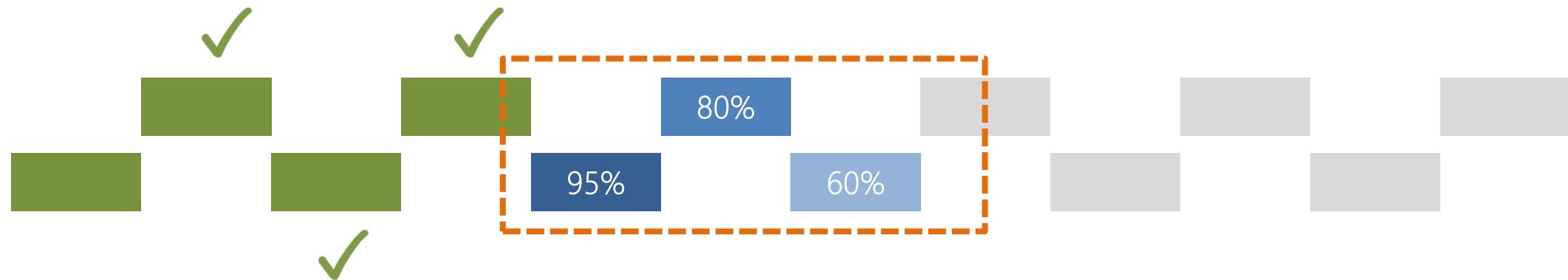
# Planning High-Level



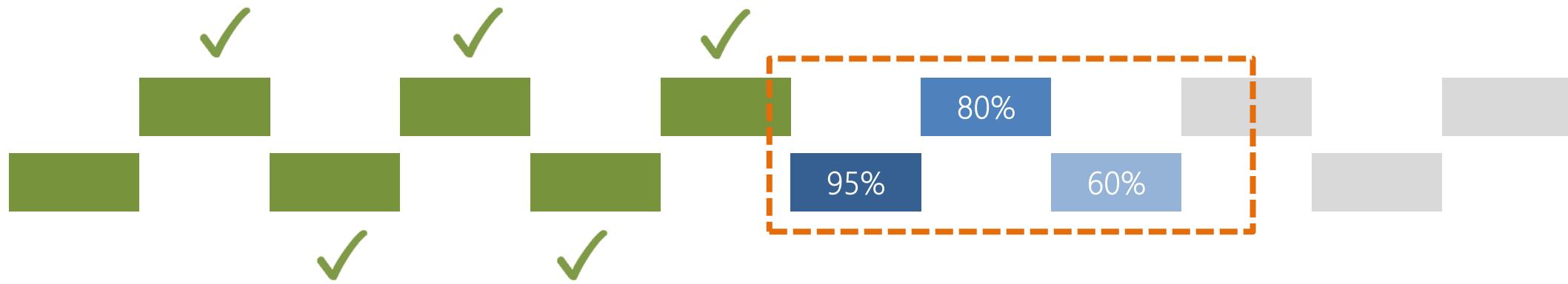
# Three Sprint Plan



# Three Sprint Plan

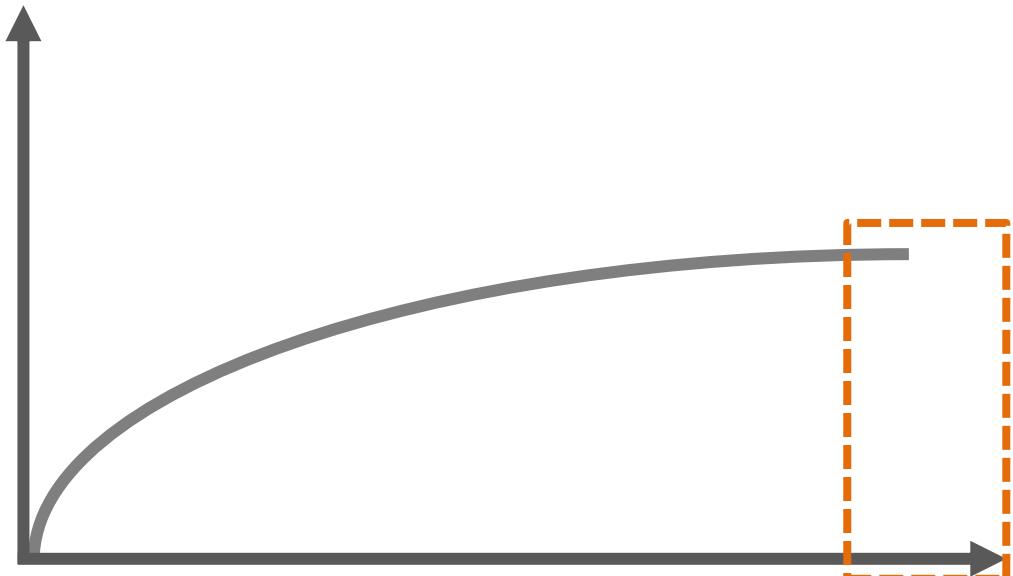


# Three Sprint Plan

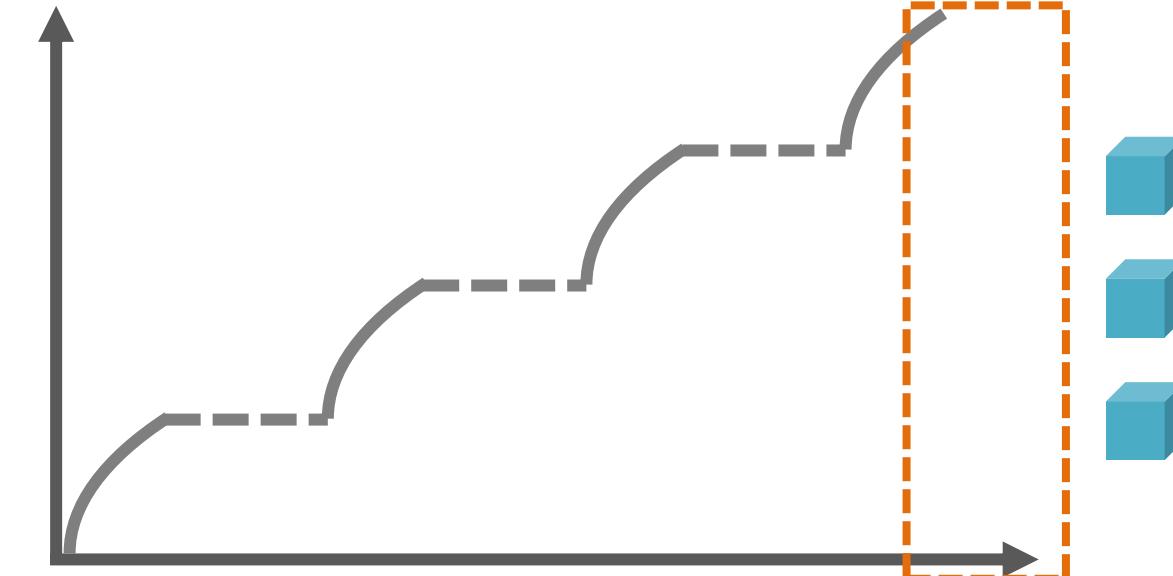


# Planning and Learning

Before

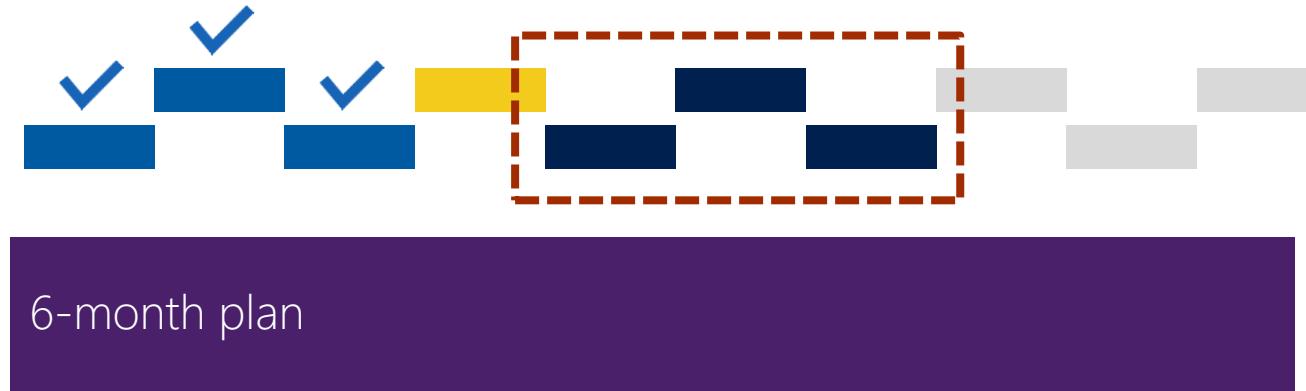


After



This approach maximizes our ability to  
learn... as we continually plan.

# Quarterly Feature Team Chats



Each team comes in and reviews with leadership three things:

1. What is the plan for the next 3-sprints?
2. Is the team healthy?
3. Any risks or issues to highlight?

# Benefits

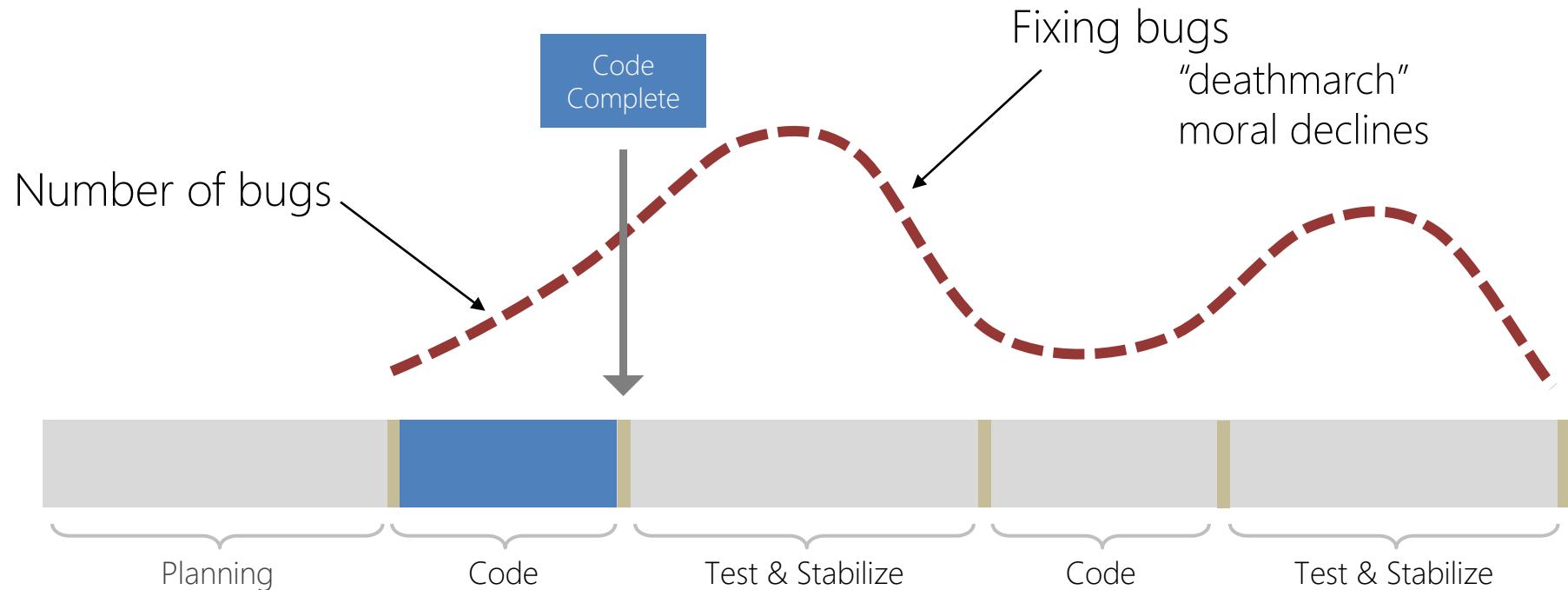
- Teams **continually plan** and learn
- Alignment is **built-in**
- Planning is driven by continual learning with
  - telemetry data on usage
  - customer feedback
  - “failing fast” through incremental execution and delivery
- **Evaluate progress**
- Ability to react if we need to change course

“Just because you’re not sick, doesn’t mean you’re healthy.”

Unknown

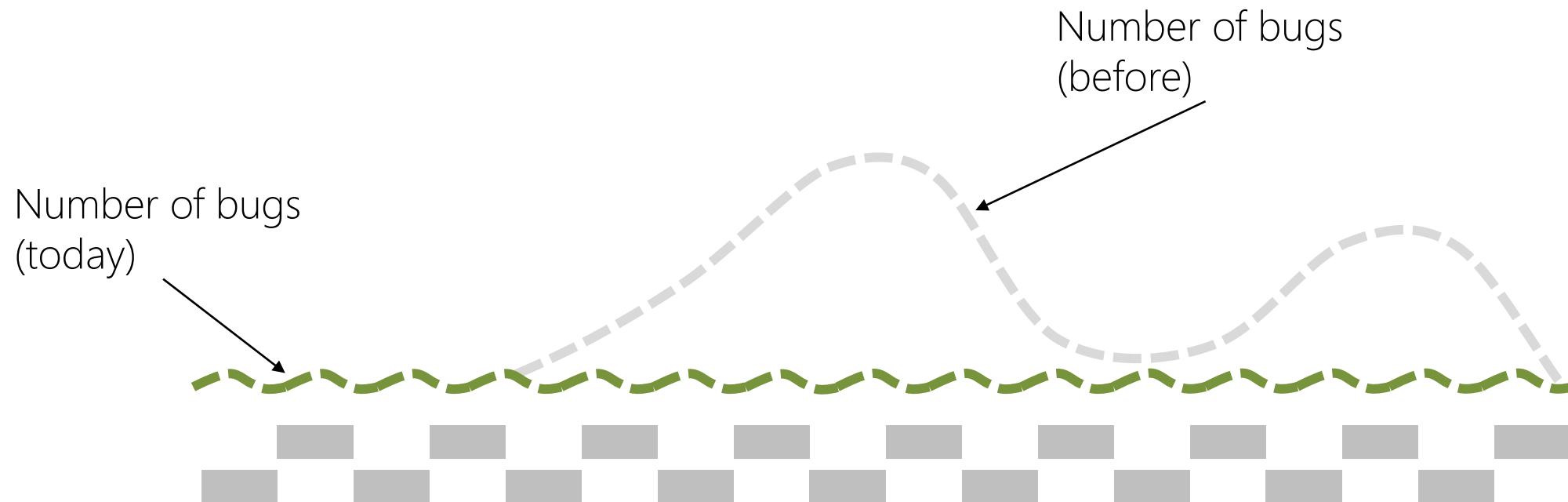
# Before

- Celebrated milestone between code phase and test & stabilize phase called code complete



# Today

- “Deathmarch” a thing of the past
- Higher team morale



# Bug Cap Solution

- Number of engineers on your team multiplied by five
- If the bugs exceed this result, stop feature development

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

# Bug Cap Solution

- Number of engineers on your team multiplied by five
- If the bugs exceed this result, stop feature development

$$10 \text{ engineers on your team} \times 5 \text{ bugs} = 50$$

RULE: If your bug count exceeds your bug cap, cease working on new features until you are back under the cap

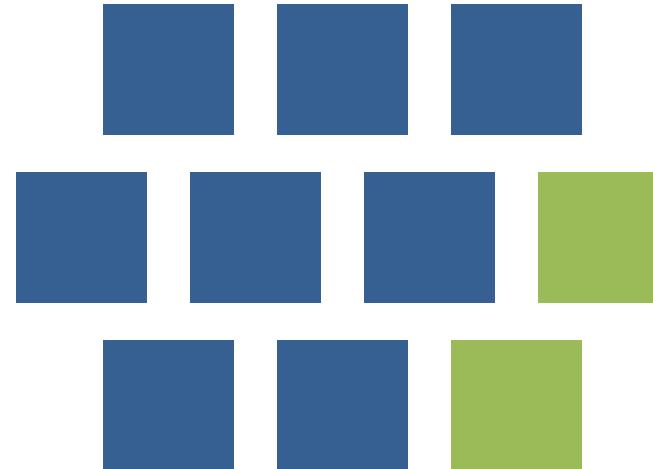
# Shielding Distractions: L-Team

- A strategy adopted by our teams to provide focus and assist with interruptions.
- The team self-organizes each sprint into two distinct sub-teams: Features and Shield

**■ F-Team**  
**(feature team)**  
Works on committed  
features (new work)

**■ L-Team**  
**(live site team)**  
Deals with all live-site  
issues and interruptions

Team of 10 Engineers



## Shielding Distractions: Goals

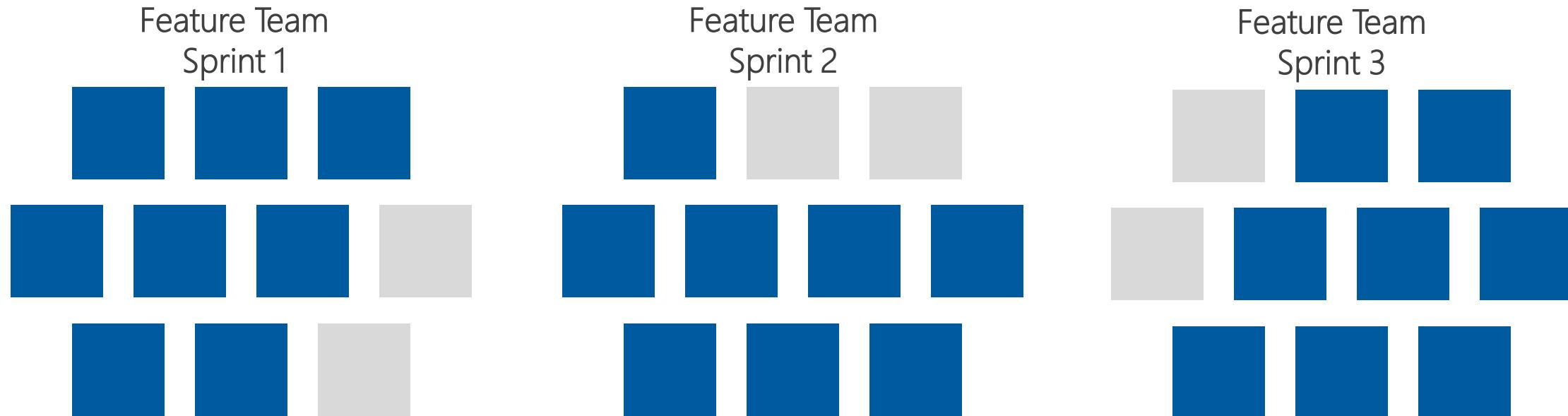
1. Allows the team to focus and commit to new work
2. Ensures people are on-call are ready to jump in and fire-fight during production issues
3. Allows the team to rotate the responsibility so you know when you are on-call

# Shielding Distractions: Rotation

Each feature team established the rotation cadence

Allows for team members to plan according with their personal life

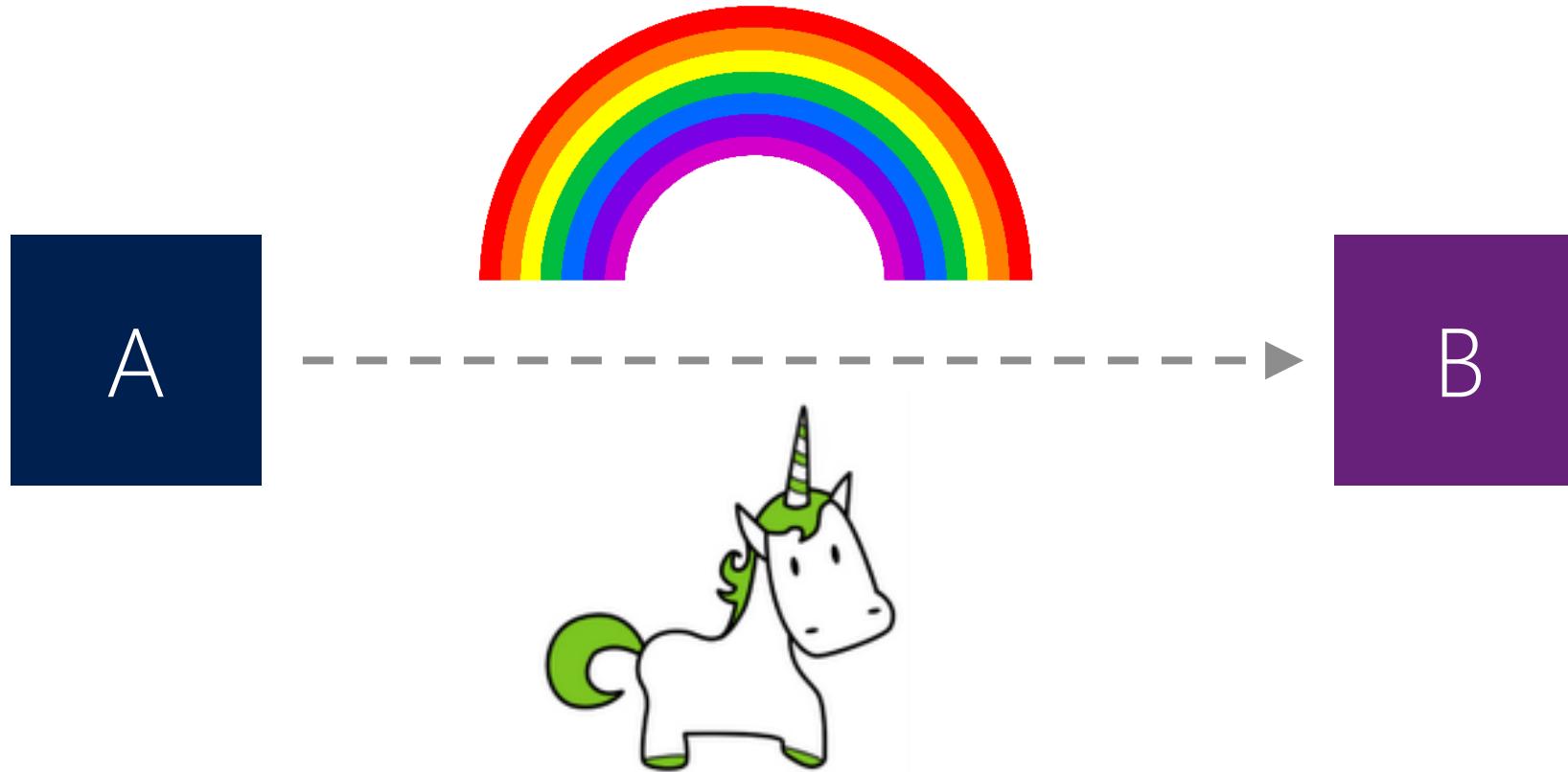
Maintain healthy work / life balance



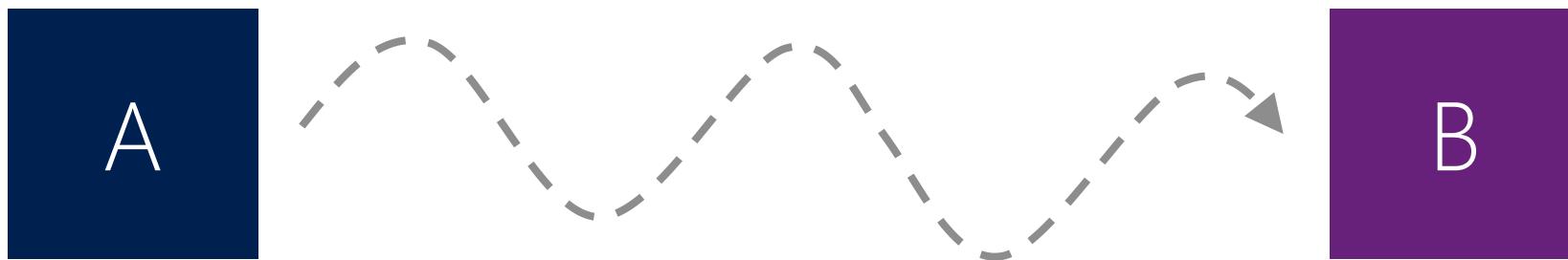
# Benefits

- Teams are **responsible** for their bug debt
- Teams work independently **at their own pace**
- Product stays **continually shippable**
- No more “deathmarches”
- Teams work at a **realistic velocity**
- Interruptions are **minimized**

# Too Good to be True?



# Continued Improvement



Journey does not end



## Case Study – Key Take-a-ways

1. Take the science seriously ... but don't be overly prescriptive
2. Stop celebrating activity ... start celebrating results
3. You can't cheat shipping
4. Build the change you want to see... you will get the behavior you are after

# Work as an Engineer

A “day in the life” of an Engineer

# A “Day in the life of” an Engineer

- Work in the Main branch
- “Walk the walk”
- Continuous Deployment
- Instrument Everything



# Issue – Integration Conflicts

The OLD way

Main Branch

Week 1

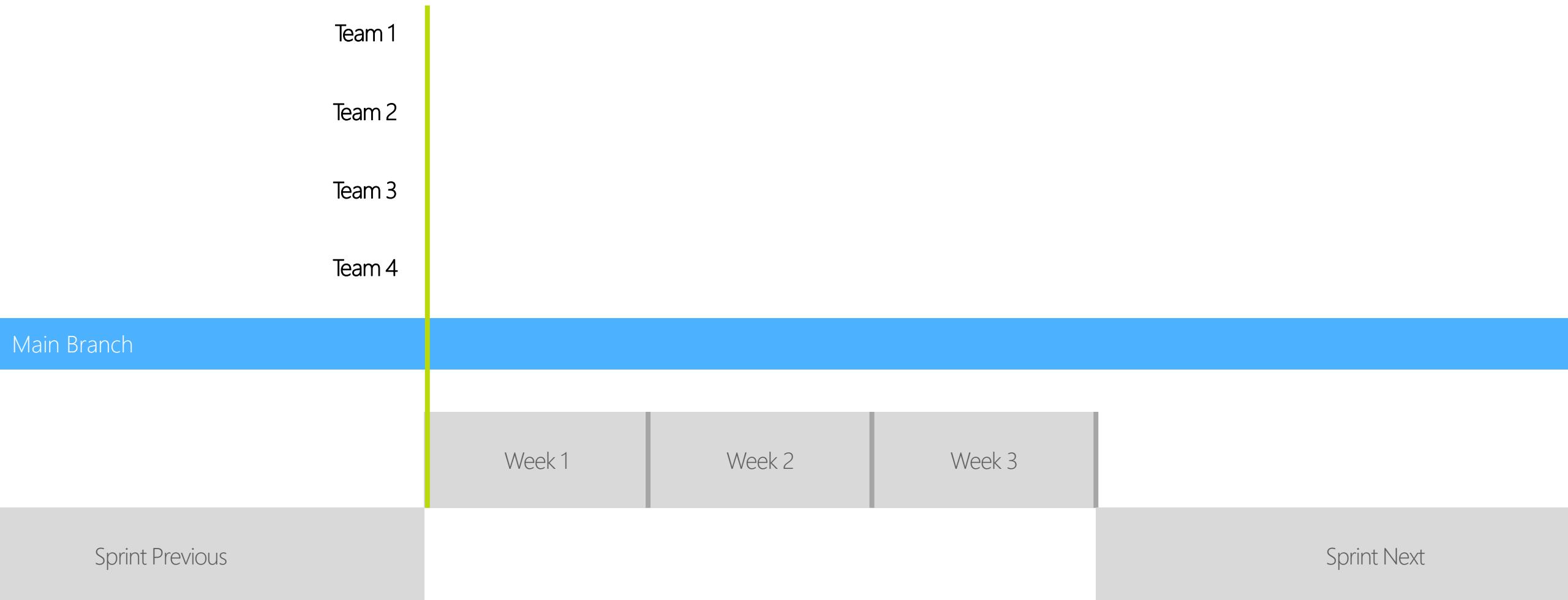
Week 2

Week 3

Sprint Previous

Sprint Next

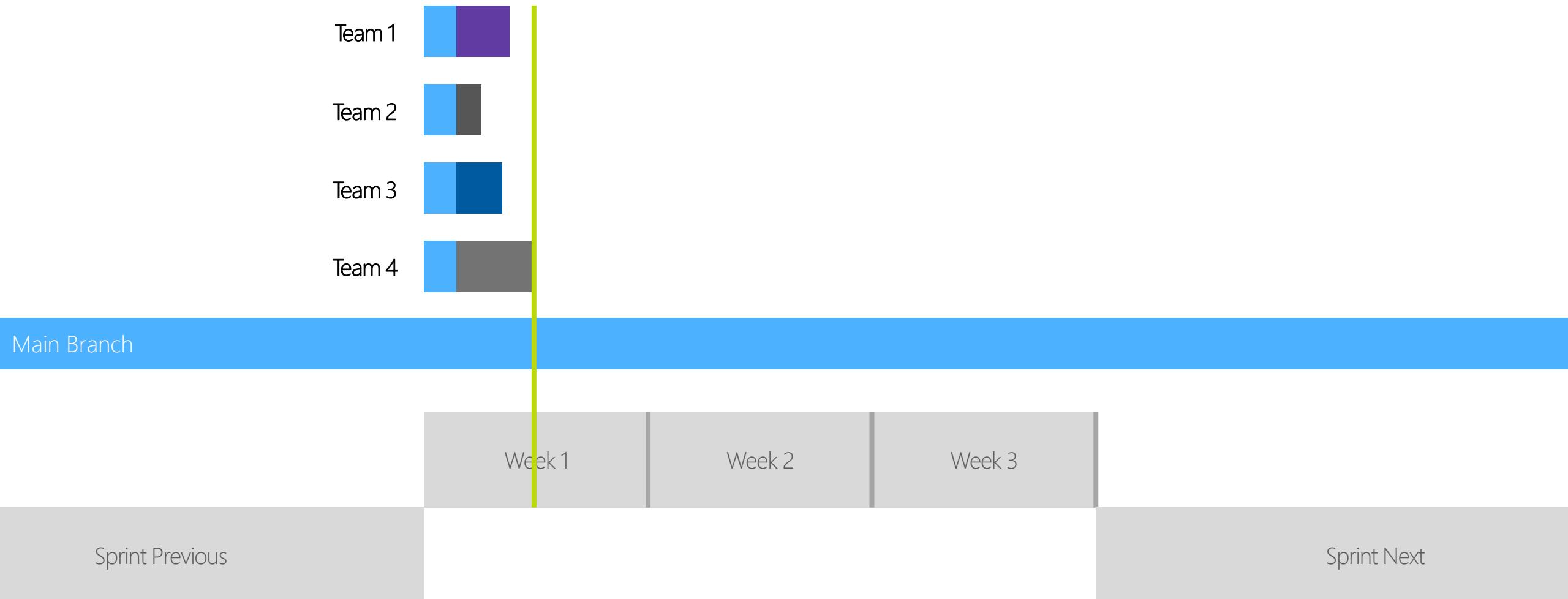
# Issue – Integration Conflicts



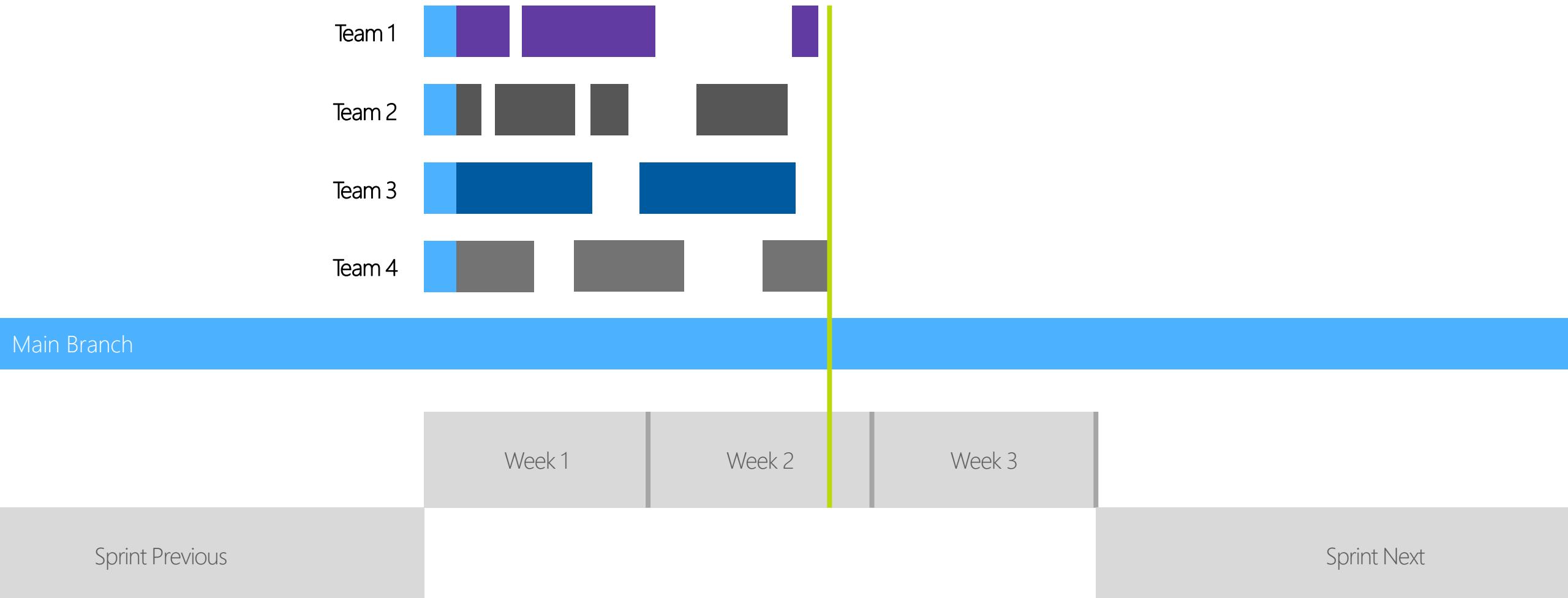
# Issue – Integration Conflicts



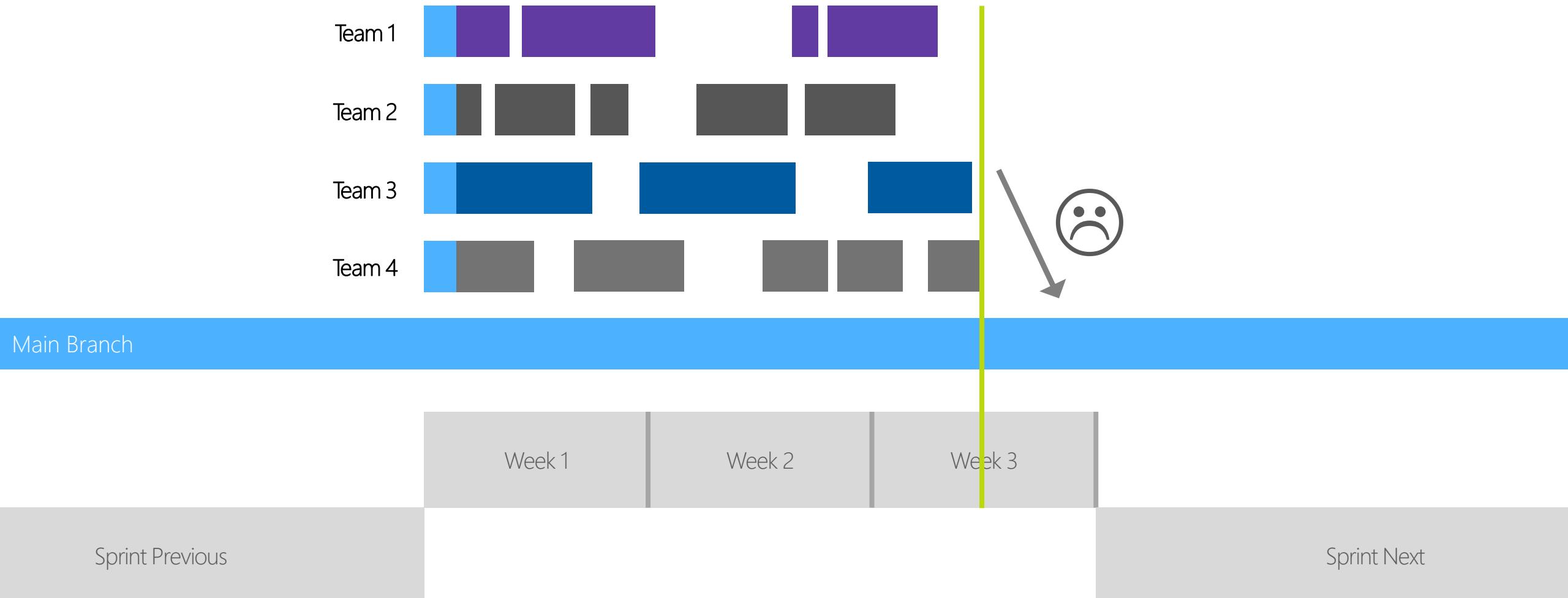
# Issue – Integration Conflicts



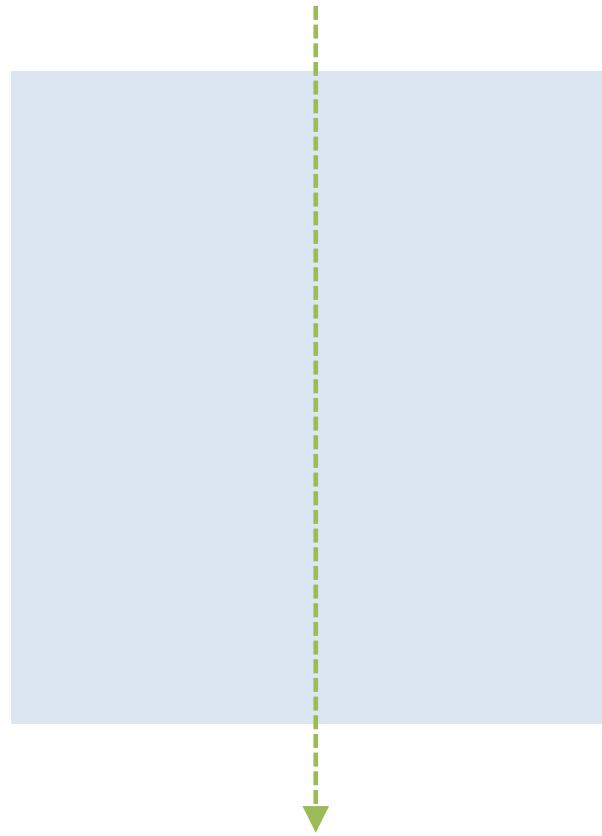
# Issue – Integration Conflicts



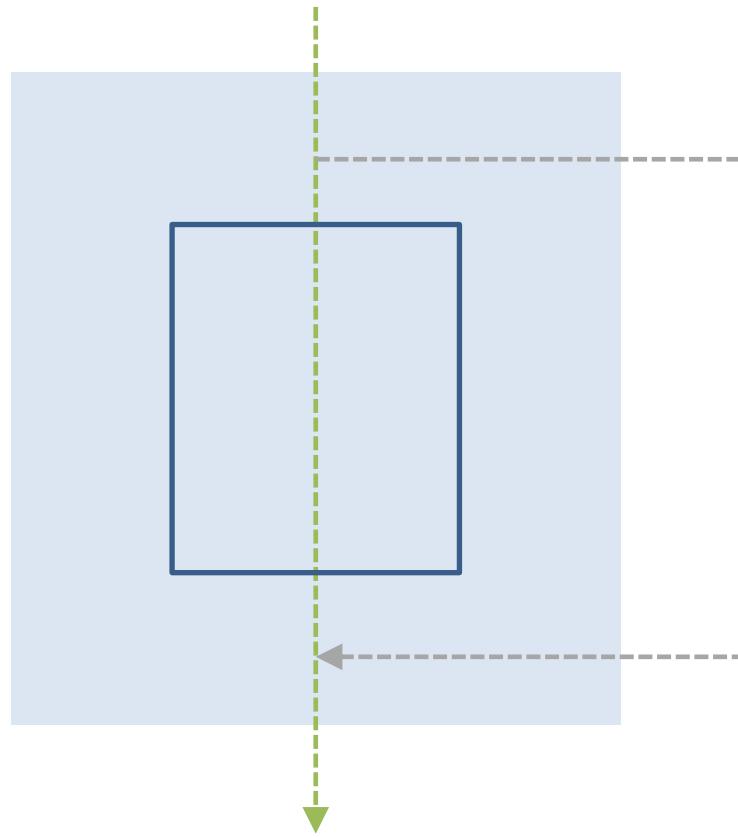
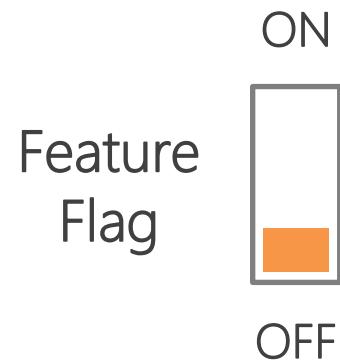
# Issue – Integration Conflicts



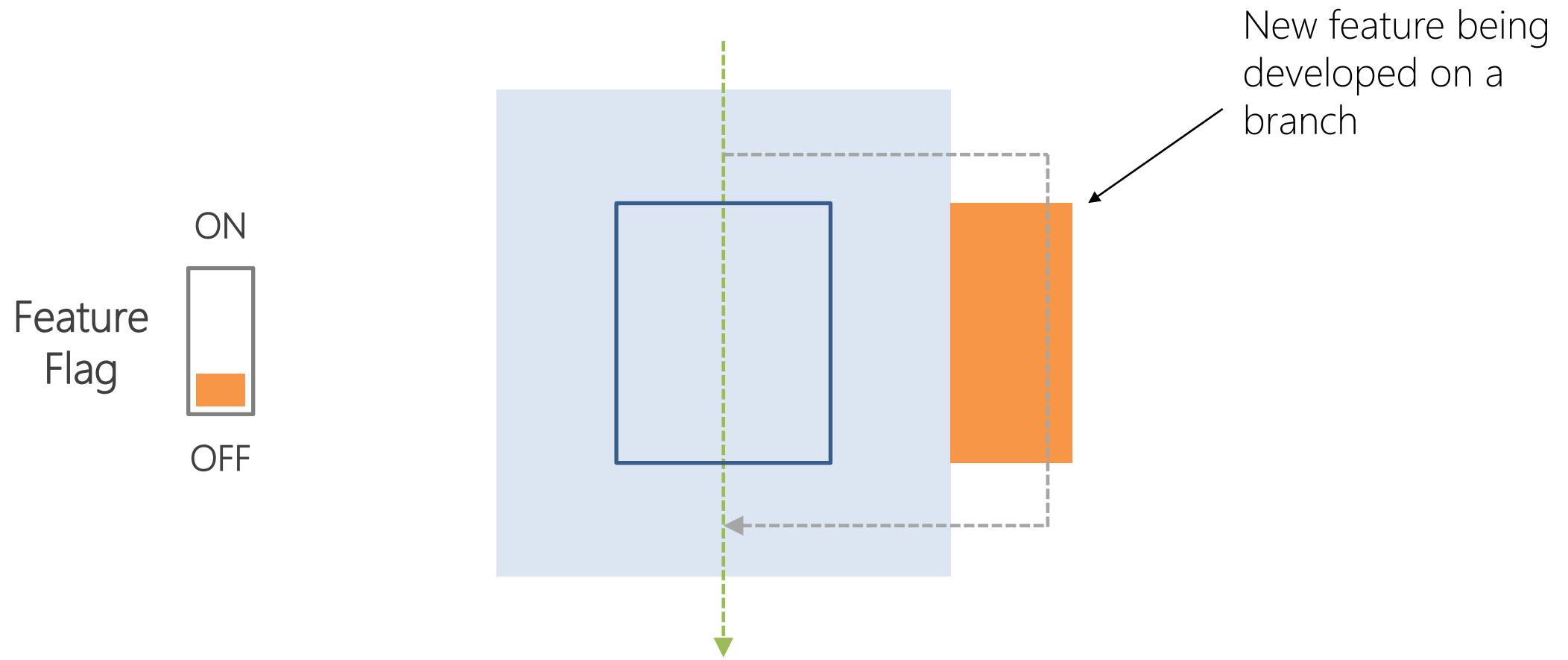
# Solution – Feature Flags



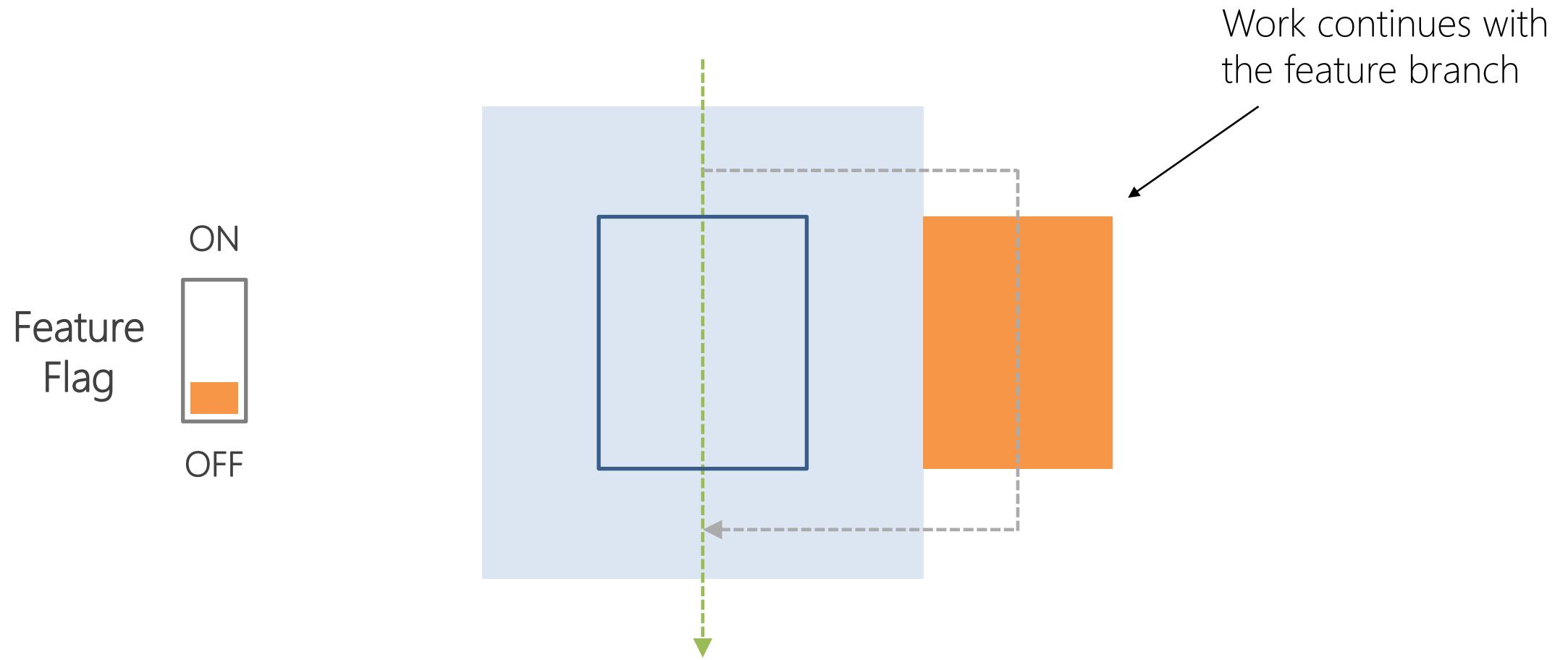
# Solution – Feature Flags



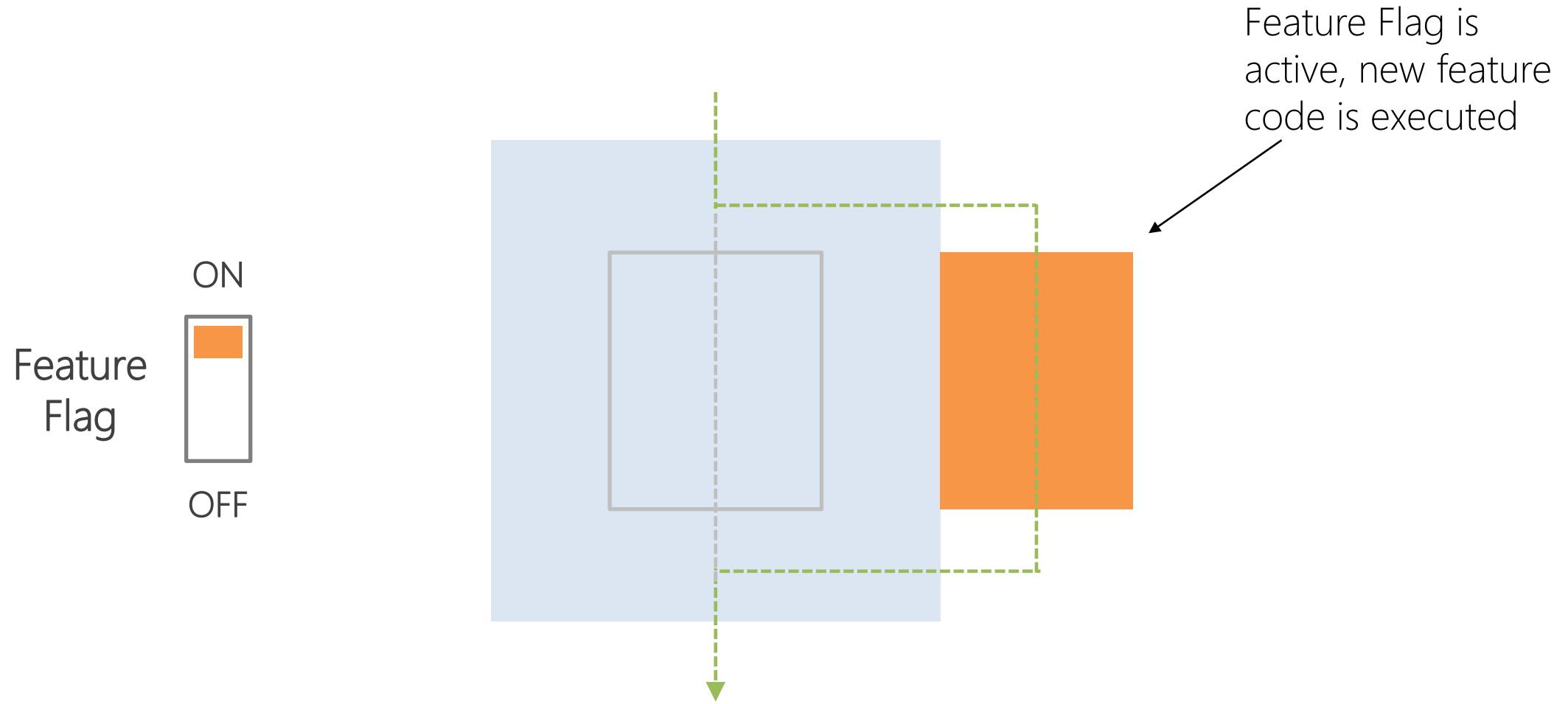
# Solution – Feature Flags



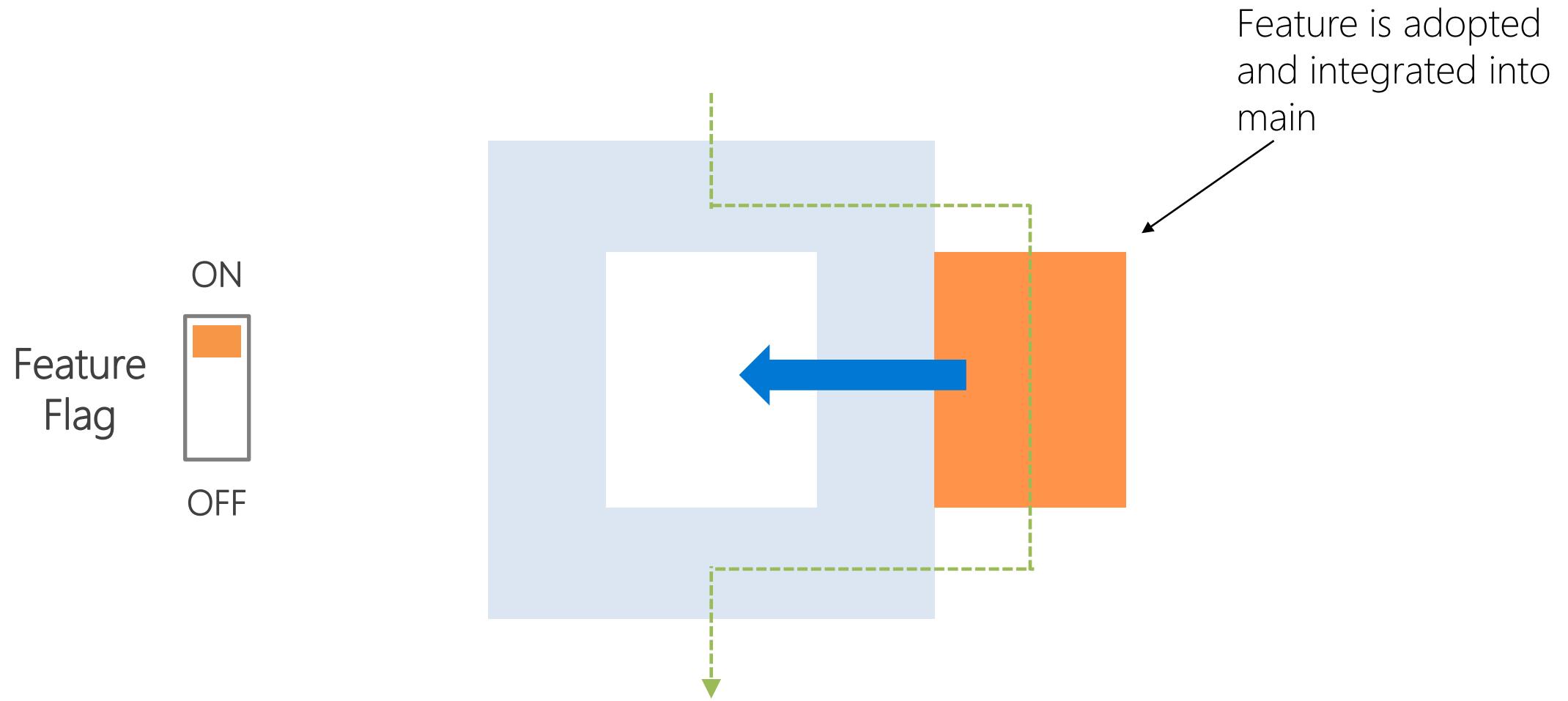
# Solution – Feature Flags



# Solution – Feature Flags

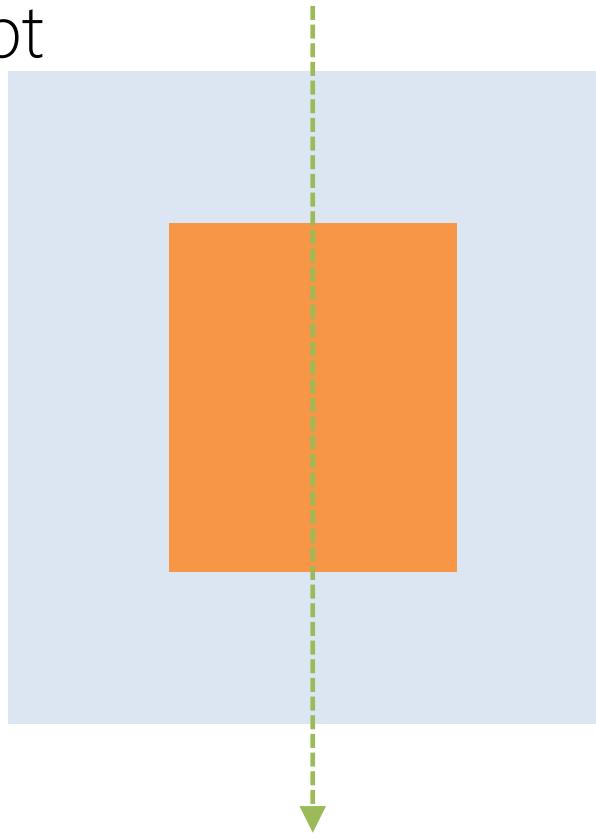


# Solution – Feature Flags



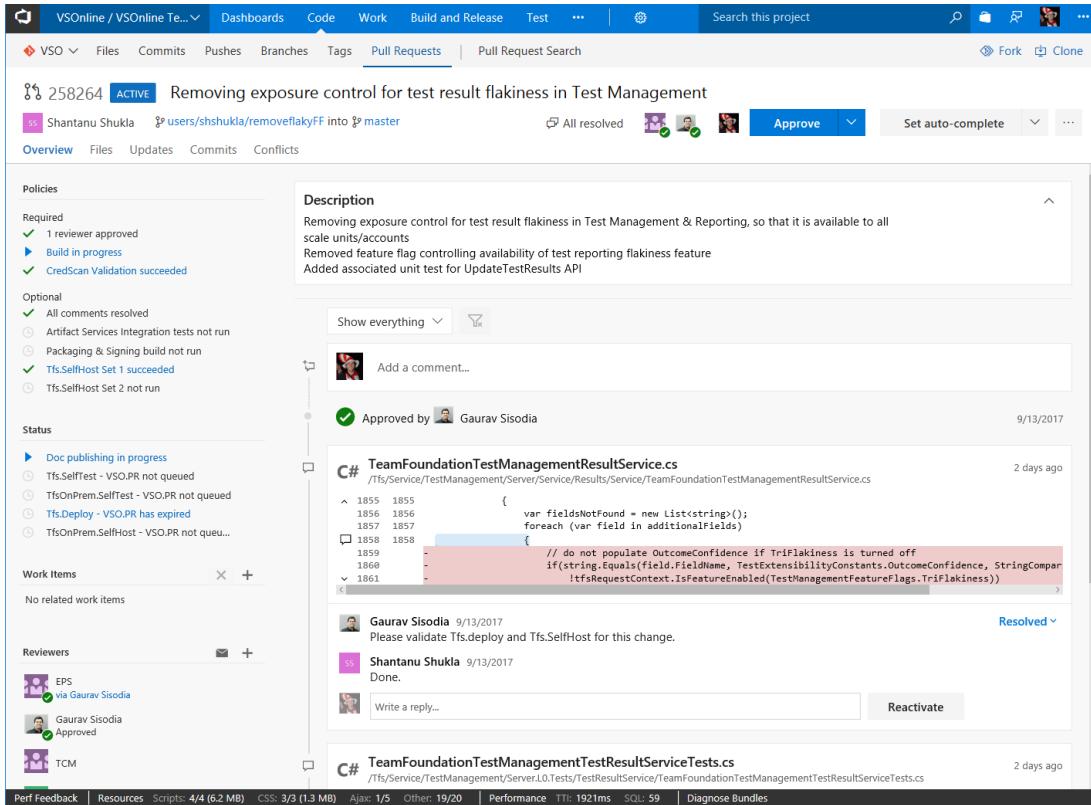
# Solution – Feature Flag

- Feature is part of application
- Mitigated integration debt



What else?

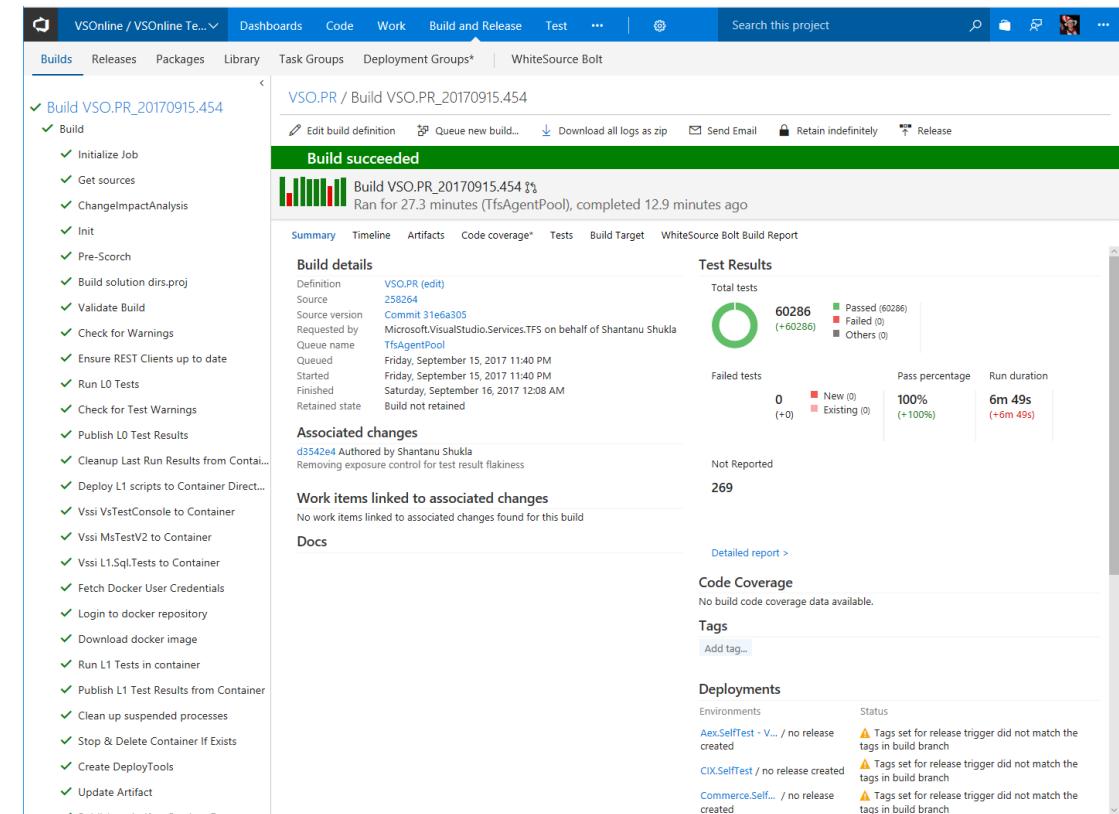
# Branch Policies



- Help keep the main branch green
- Use required reviewers
- Builds must pass
- Use security plug-ins (code scanners)

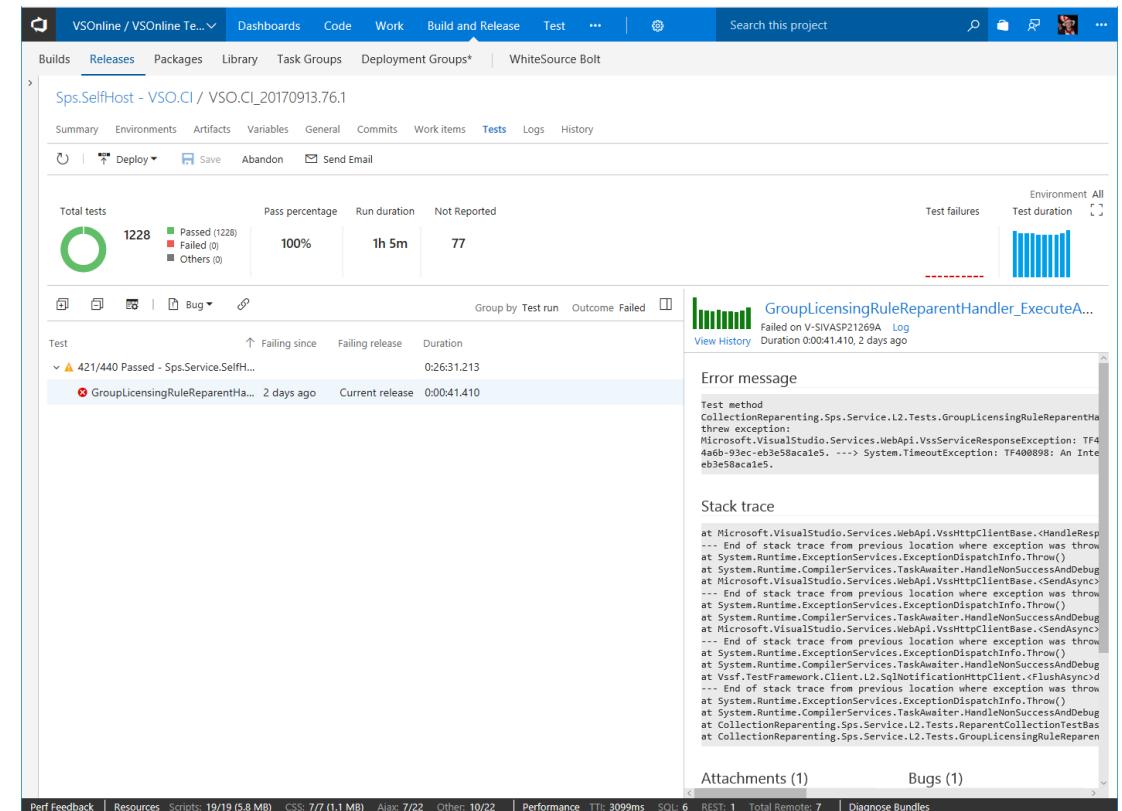
# Exercise Tests on Pull Requests

- Have fast and reliable signals
- Execute all L0 and L1 tests on a Pull Request
  - L0 – only build binaries (.exe, .dll, etc..) no dependencies
  - L1 – Use data calls and file system calls



# Exercise Tests in Continuous Integration

- Continuous Integration (CI) executes L2 tests
    - L2 – Test REST APIs
    - Test reliability is actively managed, and tests are trusted



# Quality Dashboard

- Helps with deployment decisions

# On-Call Rotation

What Service Category? (All)

What Service? Visual Studio Team Services

What Team? DRI-TFS

DRI-ServiceInsights-HolidayPeriod

DRI-SPS

DRI-SPS Extension

DRI-SPS-HolidayPeriod

DRI-TestPlanning

DRI-TestResults

DRI-TFS

DRI-TFS-HolidayPeriod

Sunday 17th

Primary: Chris Wis..., Kevin..., Kevin Mc..., Chris Wishart, Chris Sidi

Backup 1: Kevin Mc..., Chris..., Chris Wis..., Kevin McPherren, Dennis Habib

Monday 18th

Add Substitute: hover over the on call DRI name for a future time slot and click the to add substitute

Edit Substitute: hover over the substituting on call DRI name (one with a red dot) and click the to edit substitute

Remove Substitute: hover over the substituting on call DRI name (one with a red dot) and click the to remove substitute

Indicates an effective on-call substitution.

- On a feature team, the L-team (live site team) is on call
- Allows a team to work out their own rotation strategy

# Alert Fires – Responding to On-Call

**Incident Assistance Request**

**Acknowledge Request**

Assistance has been requested from you to resolve this incident:  
We need your assistance in investigating this issue as the issue is reoccurring. Please join the below bridge Join Skype Meeting <https://join.microsoft.com/meet/icmaut09/JNVGV1EV> Join by Phone Number: +14257063500 Conference ID: 5872366 Find a Local Number: <https://join.microsoft.com/dialin>

<b>46280298</b> <b>Active</b> Sev 2	Kalypso: CIA - TFS Customer Impact Monitor - [tfs][su3][scus]	<b>Start Time:</b> 2017-08-31 11:58:16
Owner: Visual Studio Team Services   DRI-Service-Delivery   v-saye		

**Description**

===== 2017-08-31 12:08:13 (PST) edited by Harshitha Pasupuleti (v-hapas) =====  
Incident Bridge for 46280298  
===== 2017-08-31 12:06:01 (PST) assigned to v-saye by Sainath Yerragudi (v-saye) =====  
Acknowledging incident  
===== 2017-08-31 12:04:55 (PST) assigned to active by Harshitha Pasupuleti (v-hapas) =====  
Engaging to look whether the issue re-occurred and received twitter escalation that experonecing from 11:44am PST:  
===== 2017-08-31 12:02:47 (PST) edited by Harshitha Pasupuleti (v-hapas) =====

We have received Kalypso alert and 43 users are impacted.

Engaging to loo into this.

===== 2017-08-31 11:58:17 (PST) submitted by connector MDM-VisualStudioOnline =====  
Resource is unhealthy.  
===== 2017-08-31 11:58:17 (PST) submitted by connector MDM-VisualStudioOnline =====

**Watchdog Name** = CIA - TFS Customer Impact Monitor, **Category** = Default

Message	Alert fired by query result monitor: CIA - TFS Customer Impact Monitor
Severity	2
ReportExpirationTime	120
Title	Kalypso: CIA - TFS Customer Impact Monitor - [tfs][su3][scus]

**Bridges**

Skype Link	Phone	Conference Id	Notes
<a href="#">Join</a>	425-706-3500	5872366	Incident Bridge for 46280298

**Diagnostics**

TSG ID:	None Specified
Component:	None Specified

**Location**

Environment:	PROD
Datacenter:	SCUS
Device Group:	None
Device Name:	su3
Slice Id:	tfs

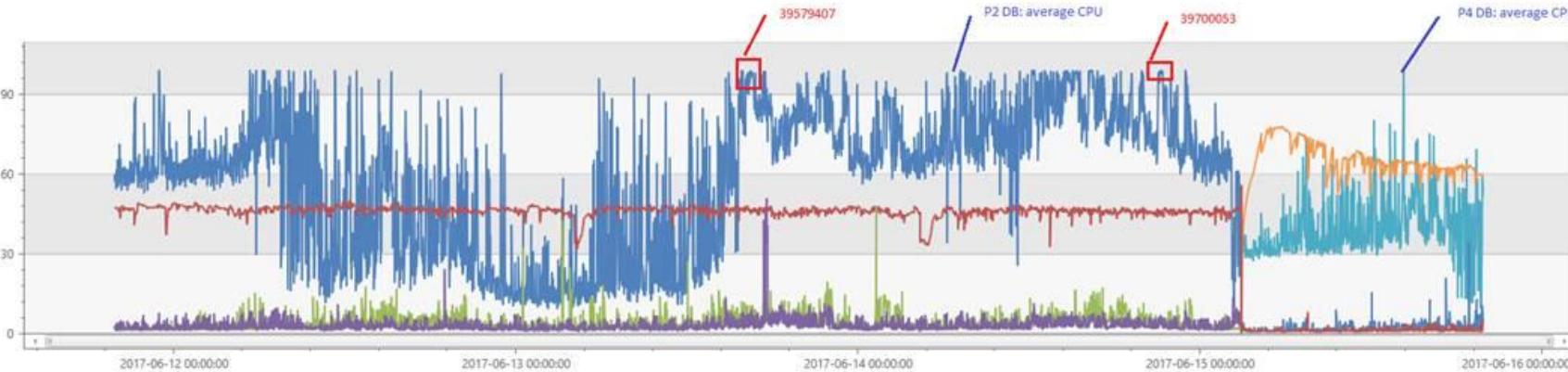
**Source**

- Acknowledge ownership
- Join bridge call
- Incident details

# Diagnosing Problems – Using Telemetry Data

```
ServiceHostAggregated
| where DatabaseName == "Tfs_tfsprodscussu5ec038584-0ece-4a4c-cc37-1cd1c92aca73"
| where Service == "tfs"
| where HostType == 4
| join (
  ActivityLog
  | where TIMESTAMP >= datetime(2017-06-06)
  | where Service == "tfs"
  | where SqlExecutionTime > 0
  | project HostId, Command, VSID, SqlExecutionCount, SqlExecutionTime) on HostId
| summarize sum(SqlExecutionTime), sum(SqlExecutionCount) by Command, VSID
| order by sum_SqlExecutionTime desc nulls last
| limit 20
```

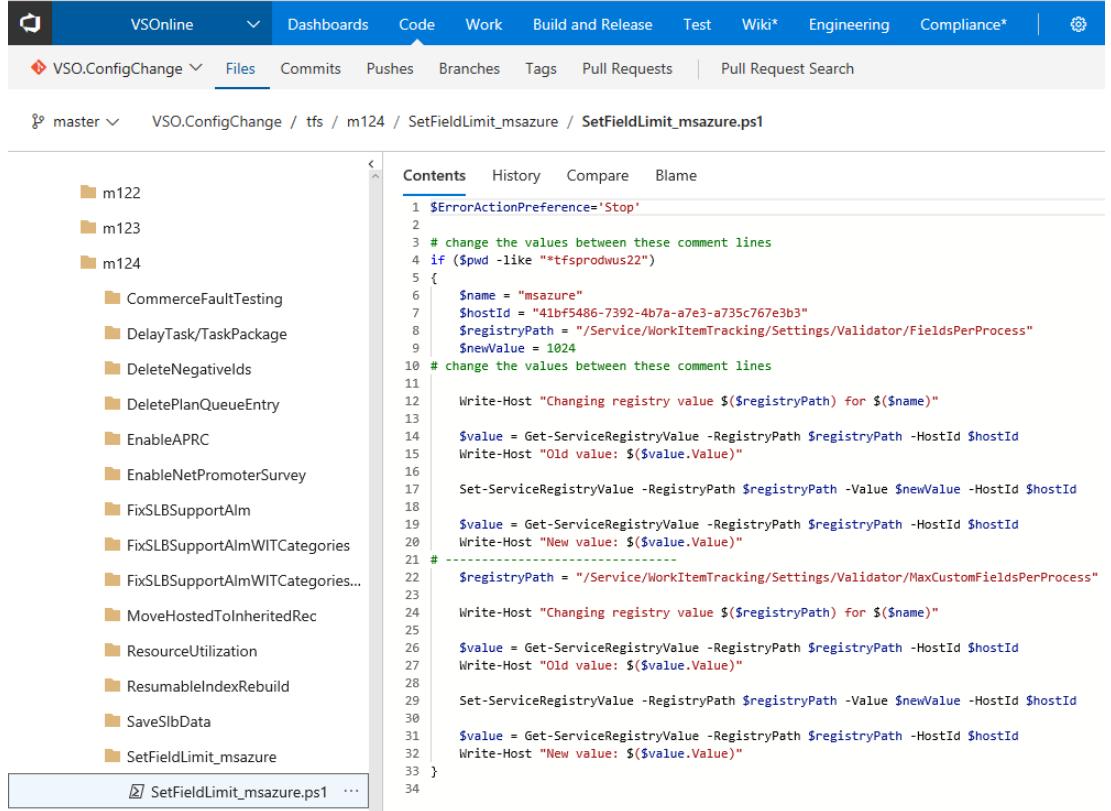
Command	VSID	sum_SqlExecutionTime	sum_SqlExecutionCount
ReportingWorkItemRevisions2.ReadReportingRevisionsGet	00000010-0000-8888-8000-000000000000	1486373133	1053286
PageWorkitemsByIds	efb6cedc-e2ad-6b24-b2ce-ad26bb898f62	124853114	31191203
QueryWorkitems	efb6cedc-e2ad-6b24-b2ce-ad26bb898f62	73883401	12370997
Containers.GetItems	a4941f8c-5d79-4de8-bb23-3e69514b0f63	66508566	21362416



- Implement telemetry early
- Instrument everything
- Application Insights (module 4)

# Configuration Changes

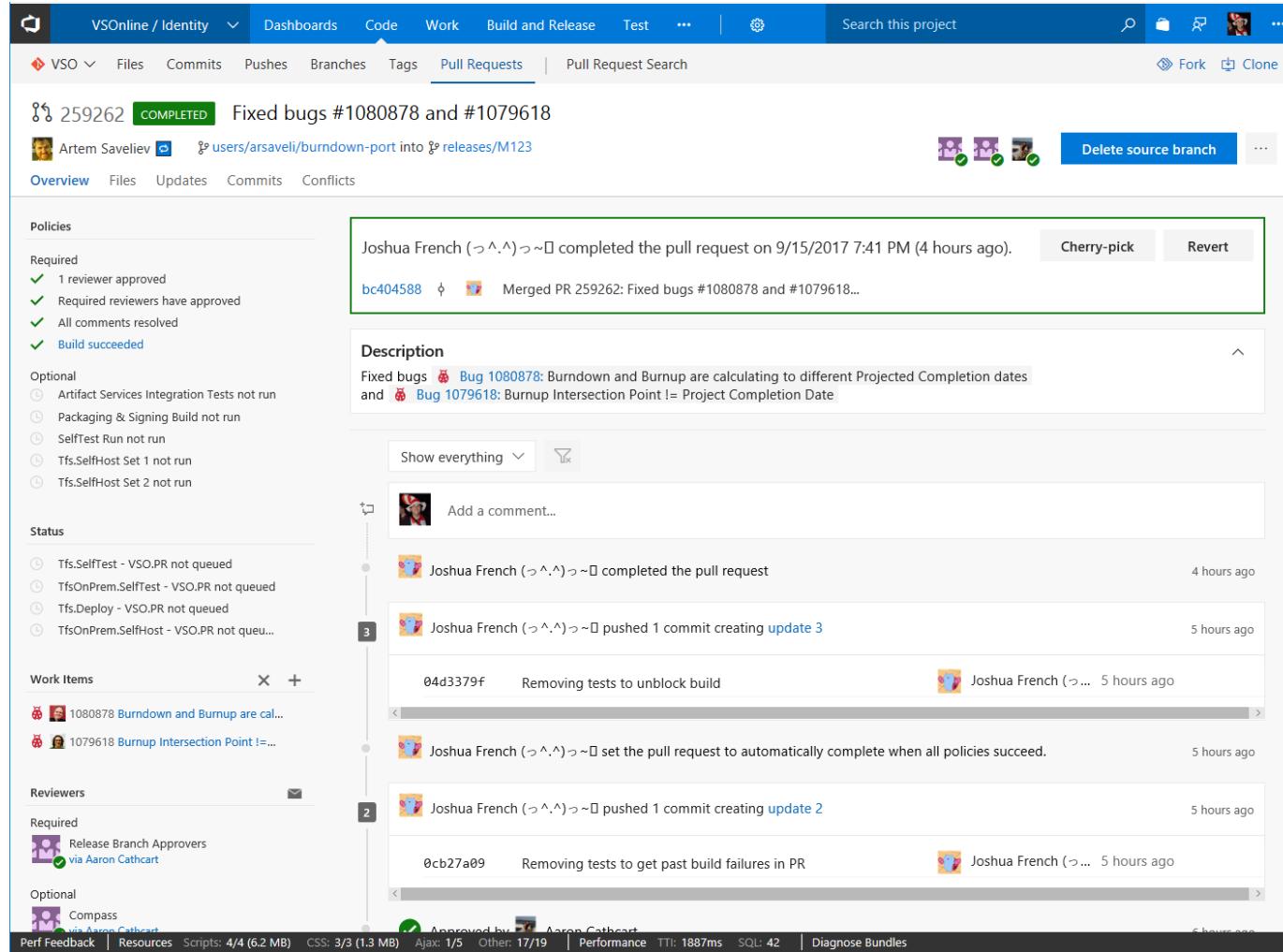
- Use scripts to implement changes
- Save in source control; allows for traceability and visibility



The screenshot shows the VSO.ConfigChange interface with the 'Files' tab selected. The left sidebar shows a file tree with several subfolders under 'm124'. The main content area displays a PowerShell script named 'SetFieldLimit\_msazure.ps1'. The script contains code to change registry values for 'FieldsPerProcess' and 'MaxCustomFieldsPerProcess' in the 'Service/WorkItemTracking/Settings/Validator' registry path. The code uses Get-ServiceRegistryValue and Set-ServiceRegistryValue cmdlets with parameters like \$registryPath, \$hostId, and \$newValue. Comment lines in the script indicate where values should be changed.

```
1 $ErrorActionPreference='Stop'
2
3 # change the values between these comment lines
4 if ($pwd -like "*tfssprodus22")
5 {
6     $name = "msazure"
7     $hostId = "41bf5486-7392-4b7a-a7e3-a735c767e3b3"
8     $registryPath = "/Service/WorkItemTracking/Settings/Validator/FieldsPerProcess"
9     $newValue = 1024
10 # change the values between these comment lines
11
12 Write-Host "Changing registry value $($registryPath) for $($name)"
13
14 $value = Get-ServiceRegistryValue -RegistryPath $registryPath -HostId $hostId
15 Write-Host "Old value: $($value.Value)"
16
17 Set-ServiceRegistryValue -RegistryPath $registryPath -Value $newValue -HostId $hostId
18
19 $value = Get-ServiceRegistryValue -RegistryPath $registryPath -HostId $hostId
20 Write-Host "New value: $($value.Value)"
21 #
22 $registryPath = "/Service/WorkItemTracking/Settings/Validator/MaxCustomFieldsPerProcess"
23
24 Write-Host "Changing registry value $($registryPath) for $($name)"
25
26 $value = Get-ServiceRegistryValue -RegistryPath $registryPath -HostId $hostId
27 Write-Host "Old value: $($value.Value)"
28
29 Set-ServiceRegistryValue -RegistryPath $registryPath -Value $newValue -HostId $hostId
30
31 $value = Get-ServiceRegistryValue -RegistryPath $registryPath -HostId $hostId
32 Write-Host "New value: $($value.Value)"
33 }
34 }
```

# Decentralize Approvers

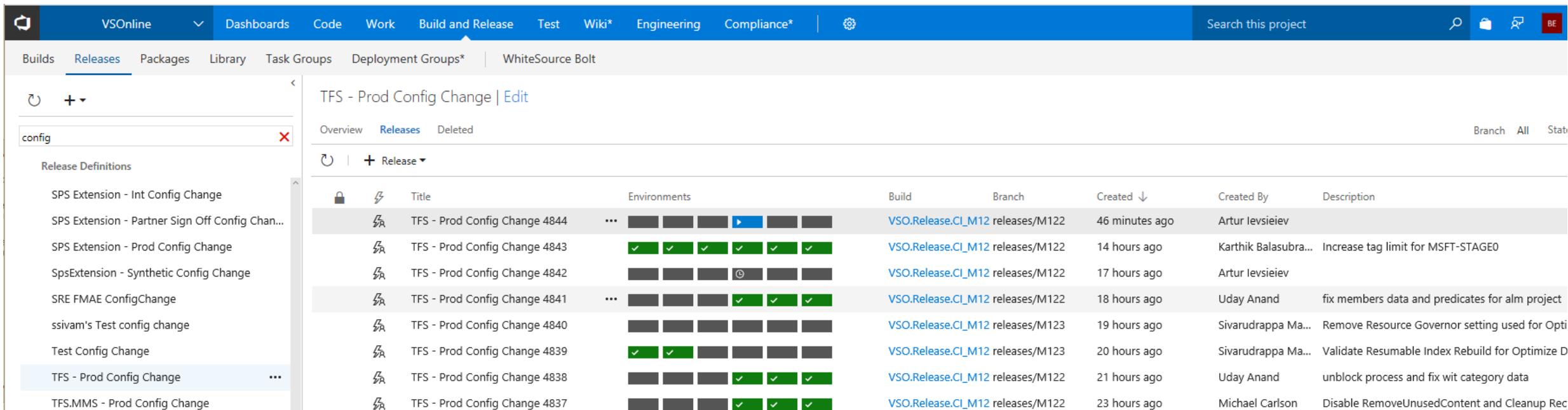


A screenshot of a pull request page in VSO Online. The pull request has been completed (COMPLETED status). The title is "Fixed bugs #1080878 and #1079618". The author is Artem Saveliev, and the merge commit is bc404588. The description notes that Joshua French completed the pull request on 9/15/2017 7:41 PM (4 hours ago). The pull request has been merged into the 'releases/M123' branch. The review status shows 1 reviewer approved, required reviewers approved, all comments resolved, and build succeeded. The status section shows optional tests like Artifact Services Integration Tests not run. The work items section lists two items: "1080878 Burndown and Burnup are cal..." and "1079618 Burnup Intersection Point != Project Completion Date". The reviewers section shows "Release Branch Approvers via Aaron Cathcart" with a green checkmark. The page footer includes performance metrics: Scripts: 4/4 (6.2 MB), CSS: 3/3 (1.3 MB), Ajax: 1/5, Other: 17/19, Performance: TTI: 1887ms, SQL: 42, and Diagnose Bundles.

- Trust in your processes and approvers
- Investments in Continuous Integration and Continuous Delivery pays off

# Deploy to Production

- Need to deploy bug fix to production
- Engineers can queue changes
- Trust in process and approvers



The screenshot shows the Microsoft VSTS (now Azure DevOps) interface for managing releases. The top navigation bar includes links for VSO Online, Dashboards, Code, Work, Build and Release, Test, Wiki\*, Engineering, Compliance\*, and a gear icon. A search bar is also present. The main content area is titled 'TFS - Prod Config Change | Edit' and shows a list of 'Release Definitions'. The list includes the following entries:

	Title	Environments	Build	Branch	Created	Created By	Description
1	TFS - Prod Config Change 4844	... (green)	VSO.Release.CL_M12	releases/M122	46 minutes ago	Artur Ievsieiev	
2	TFS - Prod Config Change 4843	✓ (green) ✓ (green) ✓ (green) ✓ (green) ✓ (green) ✓ (green)	VSO.Release.CL_M12	releases/M122	14 hours ago	Karthik Balasubramanian	Increase tag limit for MSFT-STAGE0
3	TFS - Prod Config Change 4842	... (grey) (grey) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M122	17 hours ago	Artur Ievsieiev	
4	TFS - Prod Config Change 4841	... (grey) (grey) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M122	18 hours ago	Uday Anand	fix members data and predicates for alm project
5	TFS - Prod Config Change 4840	... (grey) (grey) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M123	19 hours ago	Sivarudrappa M...	Remove Resource Governor setting used for Opti...
6	TFS - Prod Config Change 4839	✓ (green) ✓ (green) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M123	20 hours ago	Sivarudrappa M...	Validate Resumable Index Rebuild for Optimize D...
7	TFS - Prod Config Change 4838	... (grey) (grey) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M122	21 hours ago	Uday Anand	unlock process and fix wit category data
8	TFS - Prod Config Change 4837	... (grey) (grey) (grey) (grey) (grey) (grey)	VSO.Release.CL_M12	releases/M122	23 hours ago	Michael Carlson	Disable RemoveUnusedContent and Cleanup Rec...

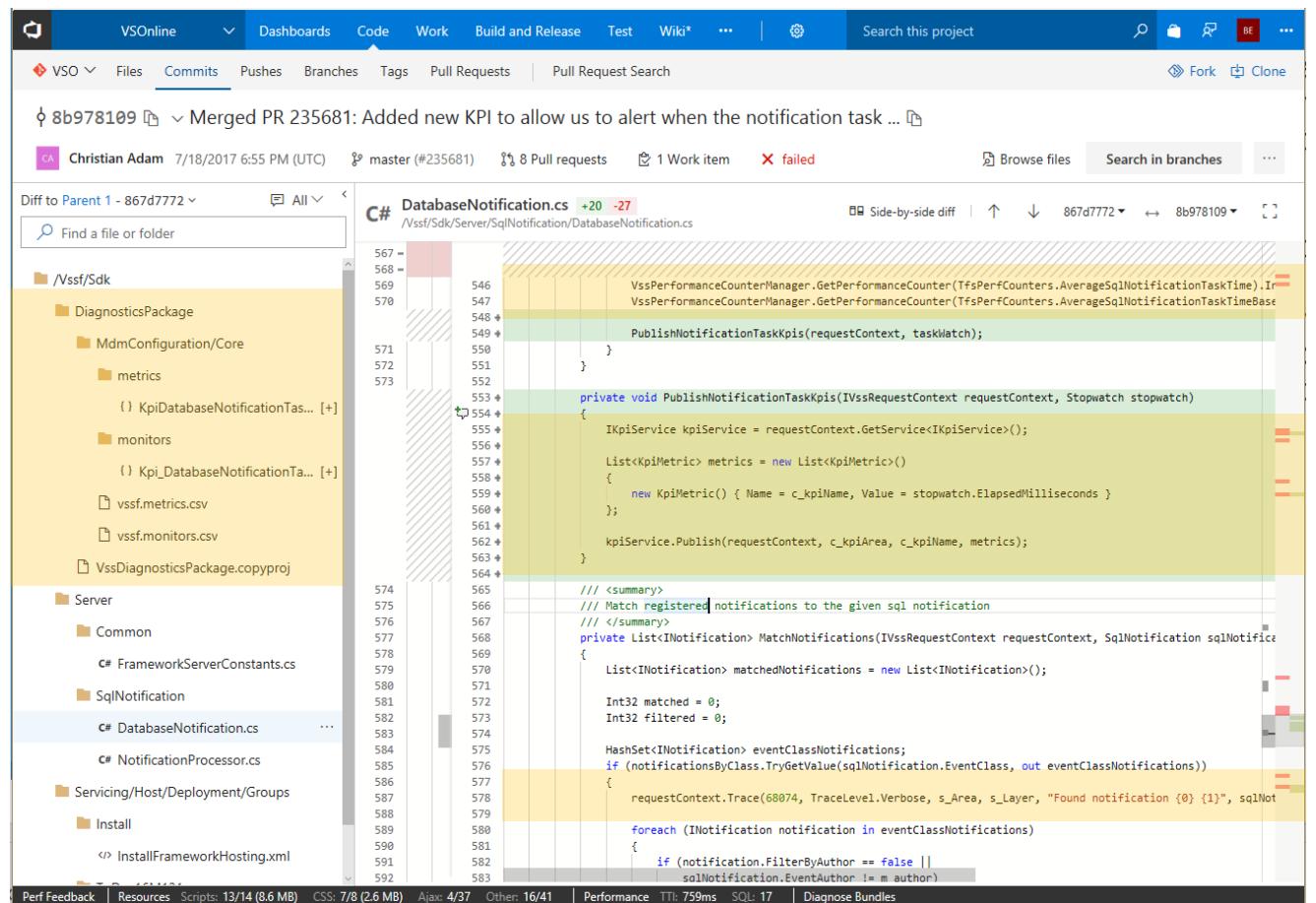
# Root Cause Analysis (RCA)

- Try to not repeat mistakes (validated learning)
- Link work items to root cause analysis
- Repair items are backlog items for the shielding team members (L-Team)

The screenshot shows the Microsoft Postmortem tool interface. The main title is "Postmortem - ICM" with the URL "icm.ad.msft.net/imp/v3/incidents/postmortem/52392". The top navigation bar includes "ICM", "Incidents", "Outages", "On Call Lists", "Resources", "Reporting", and "Administration". Below the navigation is a "Create" button and a "Search by Incident ID or other fields" input field. The main content area is divided into several sections: "Primary Incident" (4334292), "Impact Duration" (4 hours 35 minutes), "Timeline" (listing events from "Impact Start" to "Mitigation End"), "Impact" (Customer Impact: WUS2-2 Scale Unit is down, showing a graph of affected customers), "Root Cause" (Root Cause Title: WUS22 unreschedule), "Detection and Mitigation" (Detection Source: Customer, Detection Details: VSTS CRI and Web Tests of MSAG and UserFeedback were also failing, Mitigation Steps: Below commands ran from Azure Rm to help change the ReservedIP association making cloud services public IP to tproduc22-prod (e.g. 52.183.46.30 to which rollback was resolving to), Select-AzSubscription -SubscriptionId c354a4af-4934-4b0f-97e0-3f4e22222222, Remove-AzResourceGroupAssociation -ResourceName tproduc22-staging -ServiceName tproduc22, Set-AzResourceGroupAssociation -Name tproduc22 -ServiceName tproduc22), "Repair Items" (listing three repair items: Bug ID 1054187, Bug ID 1054196, and Bug ID 1054295), and "Incidents" (listing two incidents: "TFS WUS2-2 Primgem Dashboard needs to be corrected" and "Followup on MSAG wrong DL escalation" with owner Sri Hanish Kalavala).

# Telemetry Data

- Implement performance counters
- Key Performance Indicators (KPIs)
- Code tracing



The screenshot shows a code diff between 'Parent 1' and '867d7772'. The 'DatabaseNotification.cs' file is being compared. The diff highlights several changes, particularly in the 'PublishNotificationTaskKpis' method, which now includes a 'stopwatch' parameter to calculate elapsed time.

```
Diff to Parent 1 - 867d7772
C# DatabaseNotification.cs +20 -27
/Vssf/Sdk/Server/SqlNotification/DatabaseNotification.cs

567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592

private void PublishNotificationTaskKpis(IVssRequestContext requestContext, Stopwatch stopwatch)
{
    IKpiService kpiService = requestContext.GetService<IKpiService>();

    List<KpiMetric> metrics = new List<KpiMetric>()
    {
        new KpiMetric() { Name = c_kpiName, Value = stopwatch.ElapsedMilliseconds }
    };

    kpiService.Publish(requestContext, c_kpiArea, c_kpiName, metrics);
}

/// <summary>
/// Match registered notifications to the given sql notification
/// </summary>
private List<INotification> MatchNotifications(IVssRequestContext requestContext, SqlNotification sqlNotification)
{
    List<INotification> matchedNotifications = new List<INotification>();

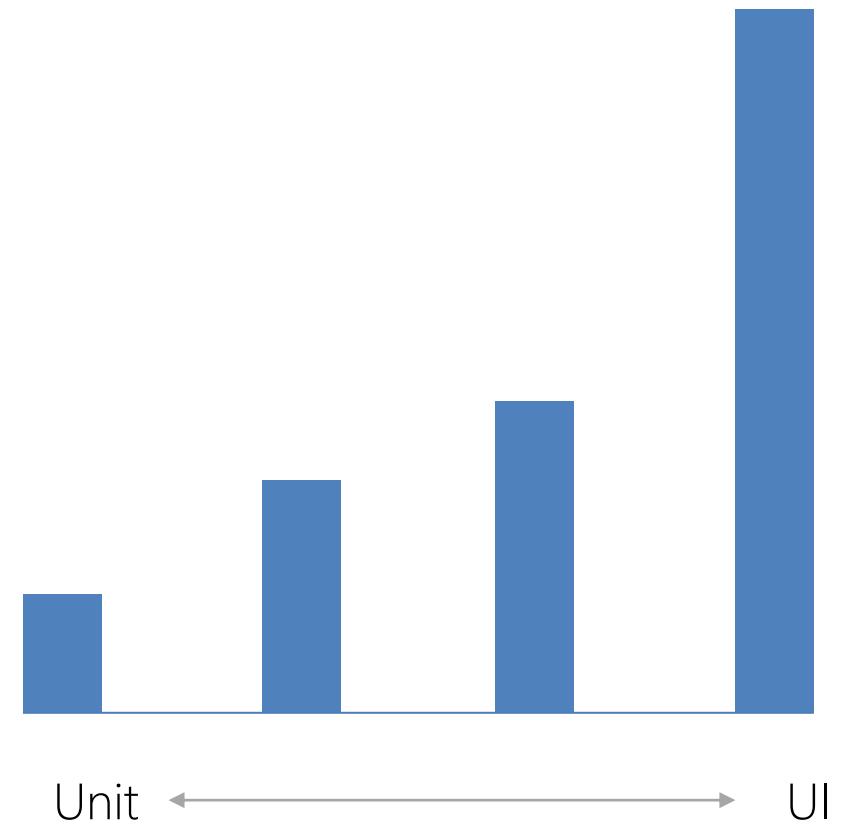
    Int32 matched = 0;
    Int32 filtered = 0;

    HashSet<INotification> eventClassNotifications;
    if (notificationsByClass.TryGetValue(sqlNotification.EventClass, out eventClassNotifications))
    {
        requestContext.Trace(68074, TraceLevel.Verbose, s_Area, s_Layer, "Found notification {0} {1}", sqlNot
        foreach (INotification notification in eventClassNotifications)
        {
            if (notification.FilterByAuthor == false ||

soNotification.EventAuthor != m_author)
```

# Test Portfolio

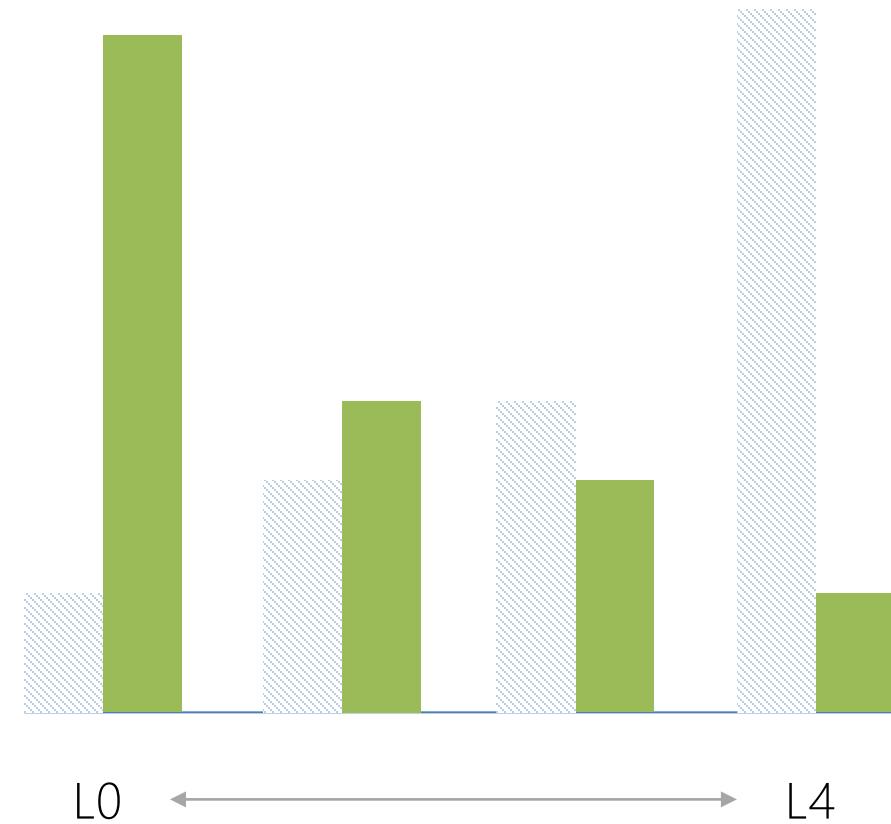
- Tests took too long to complete
- Quality was not understood for hours, sometimes days
- Tests were brittle and difficult to maintain



The OLD way

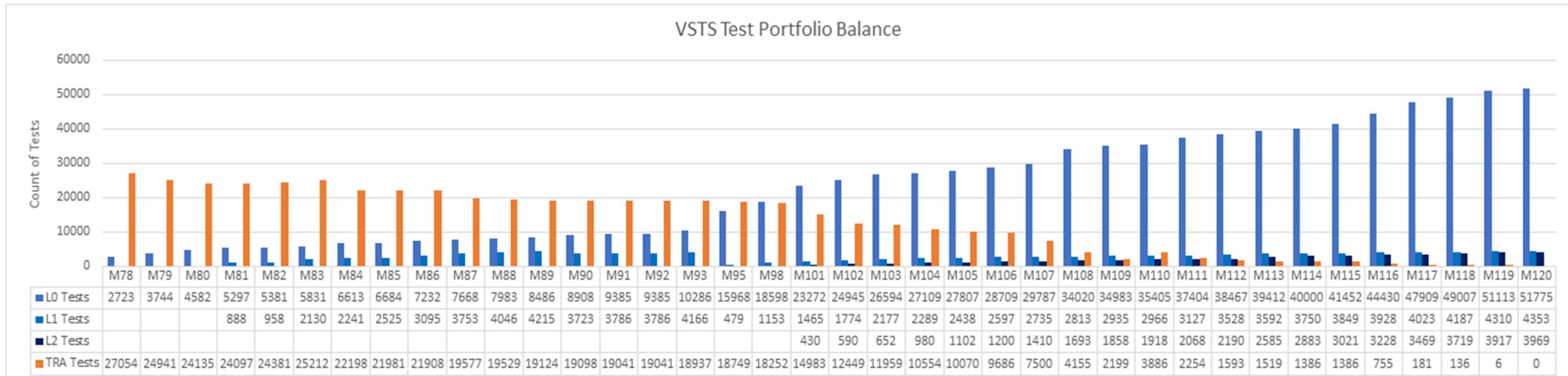
# Test Portfolio

- Tests are written at the lowest level
- Write once, run anywhere including production
- Product is designed for testability
- Test code is product code and only reliable tests survive



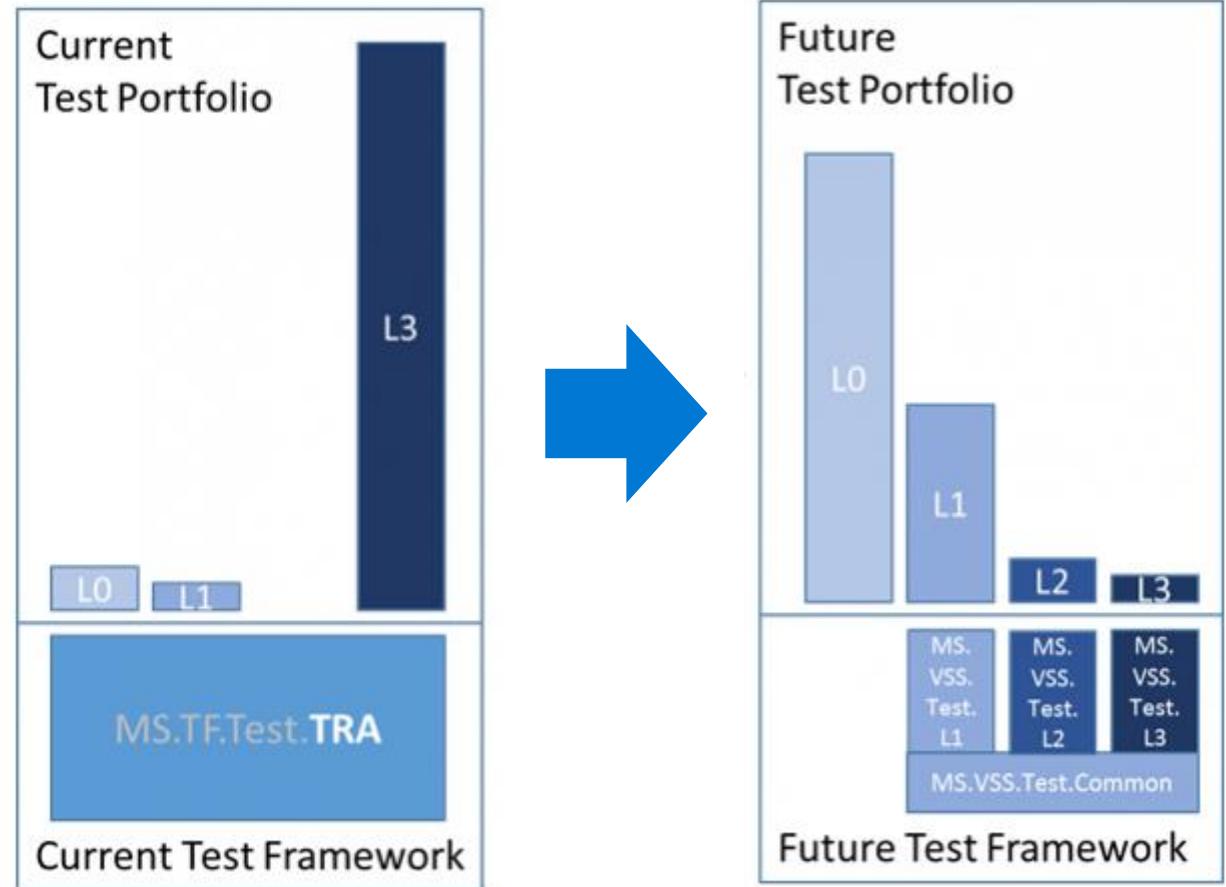
# Shift-Left from Integration to Unit Tests

- L0 - Requires only built binaries, no dependencies
- L1 - Adds ability to use SQL and file system
  - Run L0 & L1 in the pull request builds
- L2 - Test a service via REST APIs
- L3 - Full environment to test end to end



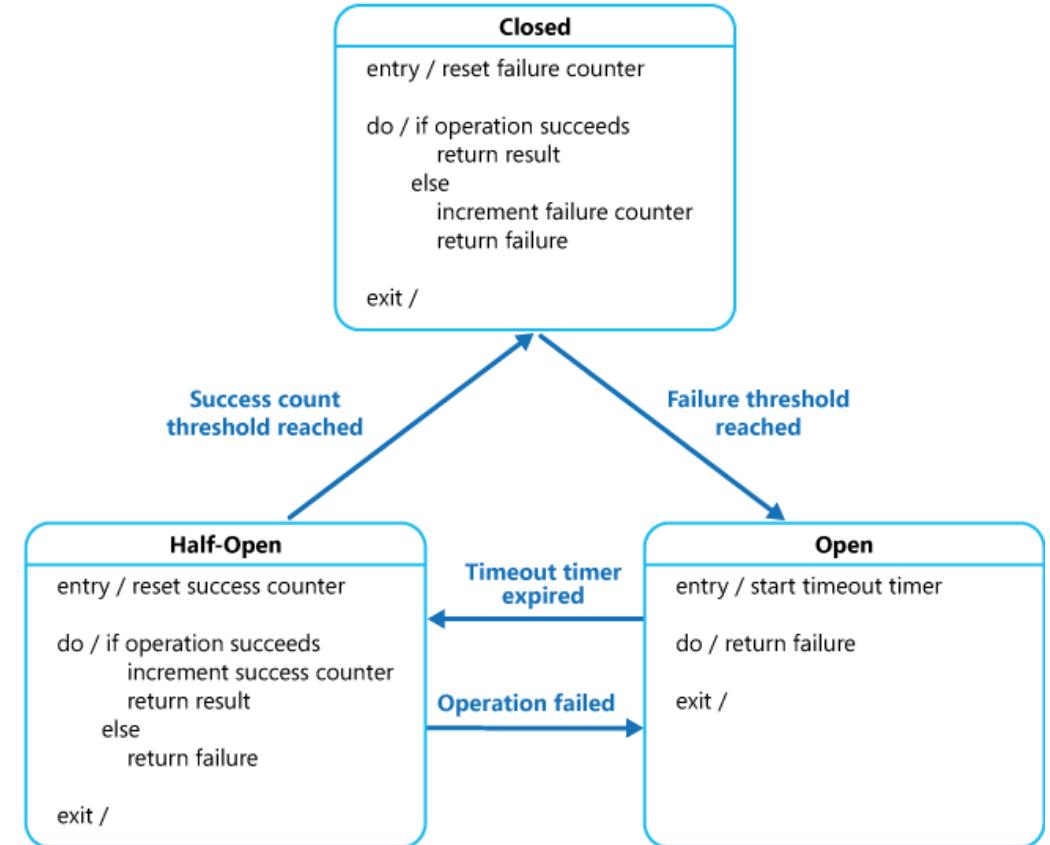
# Shift-Left from Integration to Unit Tests

- Shifted to unit tests from automated functional tests
- Core tests run before pull request
- Fast and 100% reliable build and test is critical
- Rolling tests run after commit



# Architecture

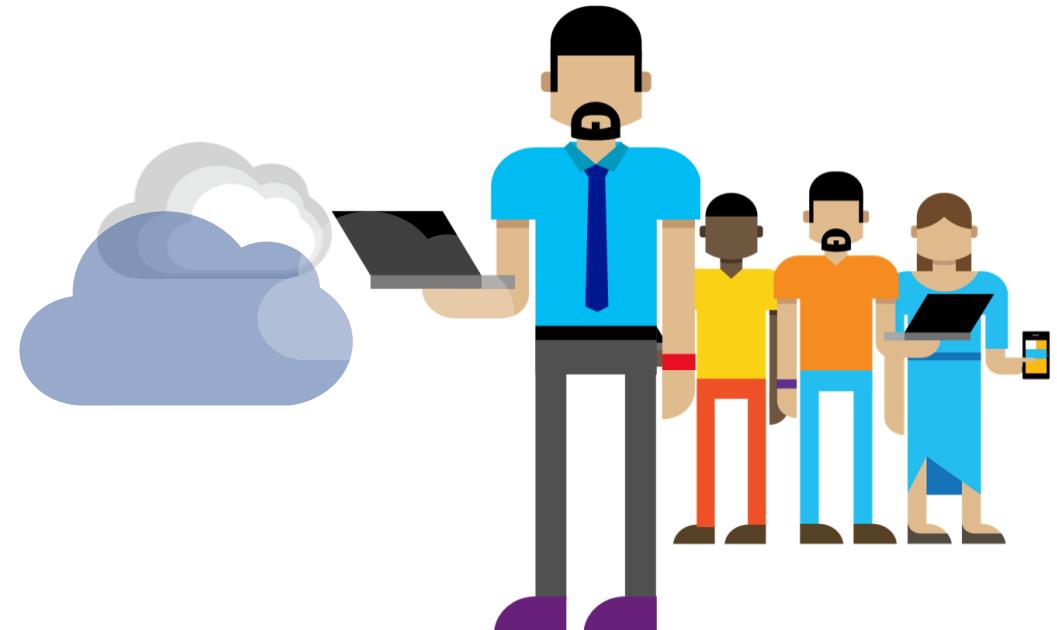
- Moved to the cloud and evolved in-flight
  - Multi-Tenant
  - Online Upgrade
  - Trace everything
- Feature flags to reduce integration debt
- Use circuit breakers and throttling to prevent cascading failure



See <https://docs.microsoft.com/en-us/azure/architecture/patterns/circuit-breaker> for the Circuit Breaker pattern.

# Security

- Throw formality aside! Make it real
- Assume the breach and plan a reaction
- Humans are human and will error



# Security Strategy

- The mindset shift to a SecDevOps culture is important in preventing breaches and assuming breaches
- Use Assuming Breaches to help answer:
  - How will we detect an attack?
  - What do we do if there is an attack or penetration?
  - How do we recover?

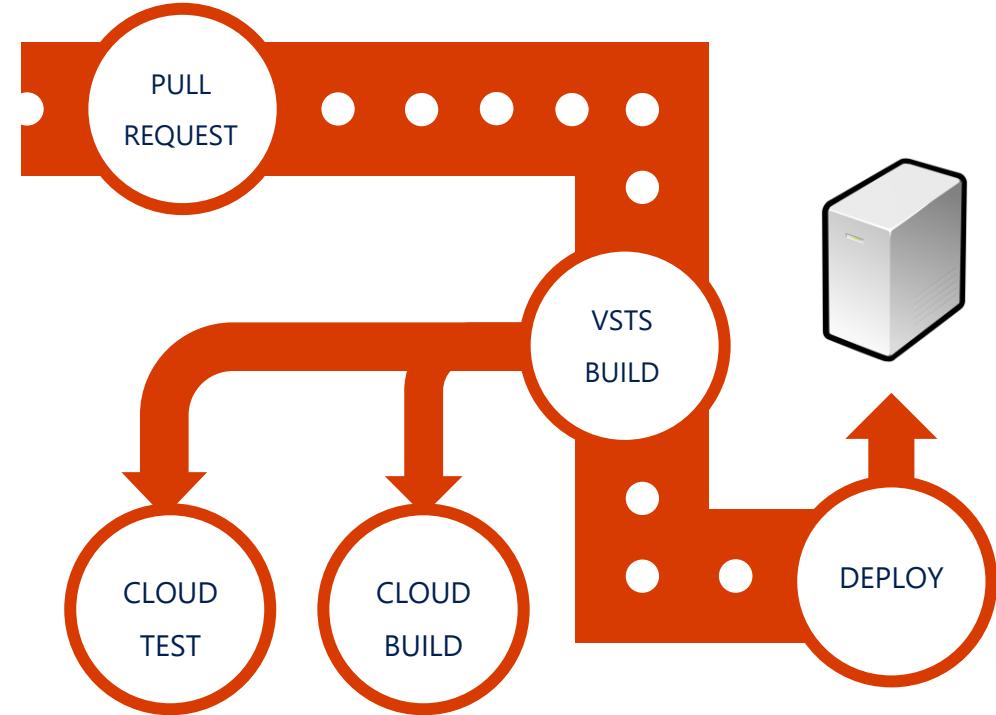


**Preventing Breaches**  
Threat Models  
Code Reviews  
Security Testing  
Software Development Lifecycle (SDLC)

**Assuming Breaches**  
War Game Exercises  
Central Security Monitors  
Live Site Penetration Tests

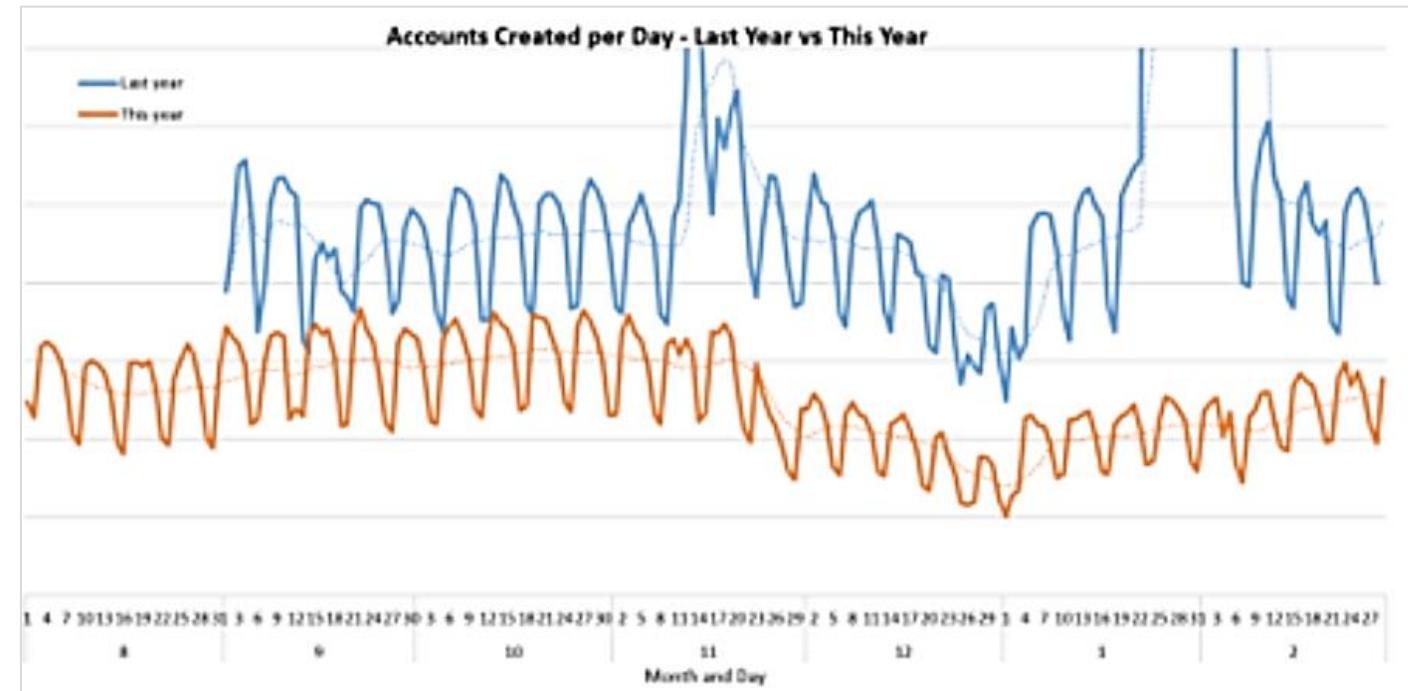
# Deployment Practices

- Deploy anyway – eat the pain
- Stay green throughout the sprint
- Follow safe deployment practices



# Running the Business on Metrics

- Start with what is most important then evolve
- Designing metrics is as hard as designing features
- Bake it into the review culture



## Work as an Engineer – Key take-a-ways

1. Feature flags reduce integration debt
2. Instrument everything
3. Bring Security in early
4. Bring Telemetry data in early
5. Shift-left and Shift-right your testing

What helped?

# Shared Customer Connection



## Customer Programs

- **On Deck champs:** customers moving to VSTS
- **MVPs:** 3<sup>rd</sup> party evangelists
- **Insiders:** customers with opinions
- **Top Customers:** largest accounts on-prem/cloud
- **Extenders/Partner program:** partners extending the TFS/VSTS core
- **DevOps Tools Sales:** our field and their customer engagements with feedback



## Customer Connections

- Events, EBCs, Site visits
- Usability studies
- **Customer development:** test hypotheses and designs with customers
- **Social media:** Stack Overflow, Twitter, DevOps blog

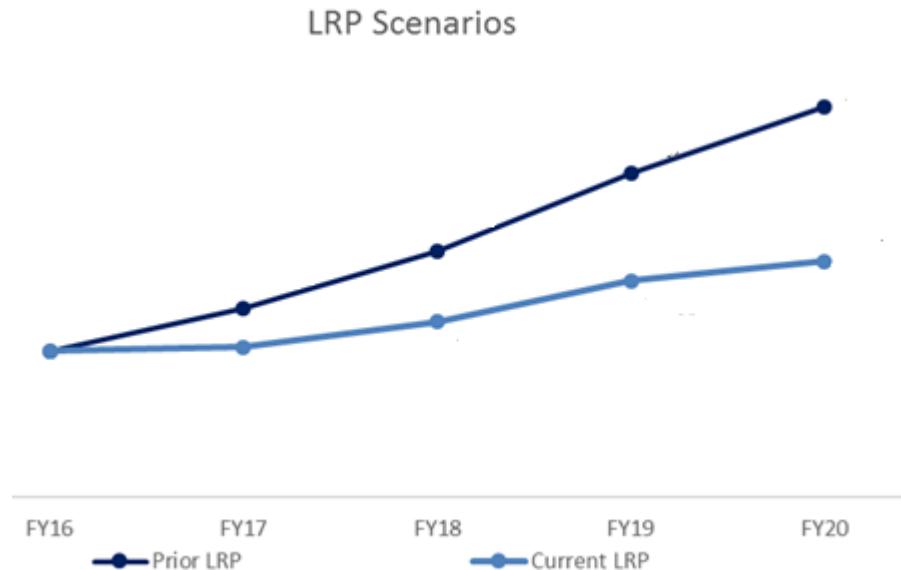


## Customer Reporting

- Customer Telemetry
- VSTS Customer Reports

# Aligning with Finance

- Long Range Planning
- Budget in July (beginning of Fiscal Year)
- Headcount, vendor count, and hardware inventories
- Continual strategy understanding & progress with Executive Vice Presidents



# Strategy Shifts: Making the Decision

- Always have more things to do
- Advocates within the team
- Investigations and conversations
- Culture changes
- Business goals once a year
- Cross-team priority spreadsheet

Initiative	Category	Stack Rank	Teams	Owner	Ending by	Starting by
Accessibility C+ and stay healthy secrets rotation	Compliance	1	all UI teams	Wendy	June/July?	in flight
Multi instance SPS	Reliability	2	platform teams	Buck	Dec?	in flight
Epic Progress	Scenarios	3	all teams	VSTS Directors	in flight	in flight
Data import	Adoption	4	platform teams	Mario	Aug	in flight
Organization support: phase 1	Scenarios	5	platform teams	Buck delegate	Sept	in flight
Containerize VSTS	Agility + COGS	8	CAT, CSF, SI, Kalypso	?	in flight	in flight
VSTS CLI	Scenarios	9	VC, RM, Build	Will Smythe	Oct	in flight
Organization support: phase 2 for social	Scenarios	10	all teams		in flight	in flight
Public Projects	Scenarios	11	all teams	Matt Cooper	Dec	Dec
Org urls for social	Scenarios	12				120
GDPR	Compliance	13				summer
National Cloud: Fedramp/Fairfax, Mooncake	Adoption	14				
High Availability Better Disaster Recovery:						
georeplication	Reliability	15			Aug	
Accessibility B	Compliance	16				
team project soft delete	Reliability	16-postponed				
auditing	Adoption	17 - postponed		Mario		
storage limits ala OneDrive	Reliability	18- postponed				
Loc	Adoption	postponed				
Fire and forget PR	Agility	postponed	EPS, Build			
Fault Injection	Reliability	postponed	all teams			

# Feed into Agile Planning: ROB (Rhythm of Business)

- Define epics and scenarios with their success metrics
- Epic Reviews
- Yellow Sticky Exercise
- Feature Chat – Hear it from the teams
- All Hands
- Sprint Mail

Leadership is responsible  
for the big picture



Teams are responsible for the detail

# Measure What's Important (KPIs)

## Usage

- Acquisition
- Engagement
- Satisfaction
- Churn
- Feature Usage

## Velocity

- Time to Build
- Time to Self Test
- Time to Deploy
- Time to Learn

## Live Site Health

- Time to Detect
- Time to Communicate
- Time to Mitigate
- Customer Impact
- Incident Prevention Items
- Aging Live Site Problems
- SLA per Customer
- Customer Support Metrics

## Engineering Scorecard - Sprint 124

### Things we don't watch

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



# Before and After

4 to 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

3-week sprints

Vertical teams

Team rooms

Continual Planning & Learning

PM & Engineering

Continual customer engagement

Everyone in main

8-12 person teams

Publicly shared roadmap

Zero debt

Specs in PPT

Open source

Flattened organization hierarchy

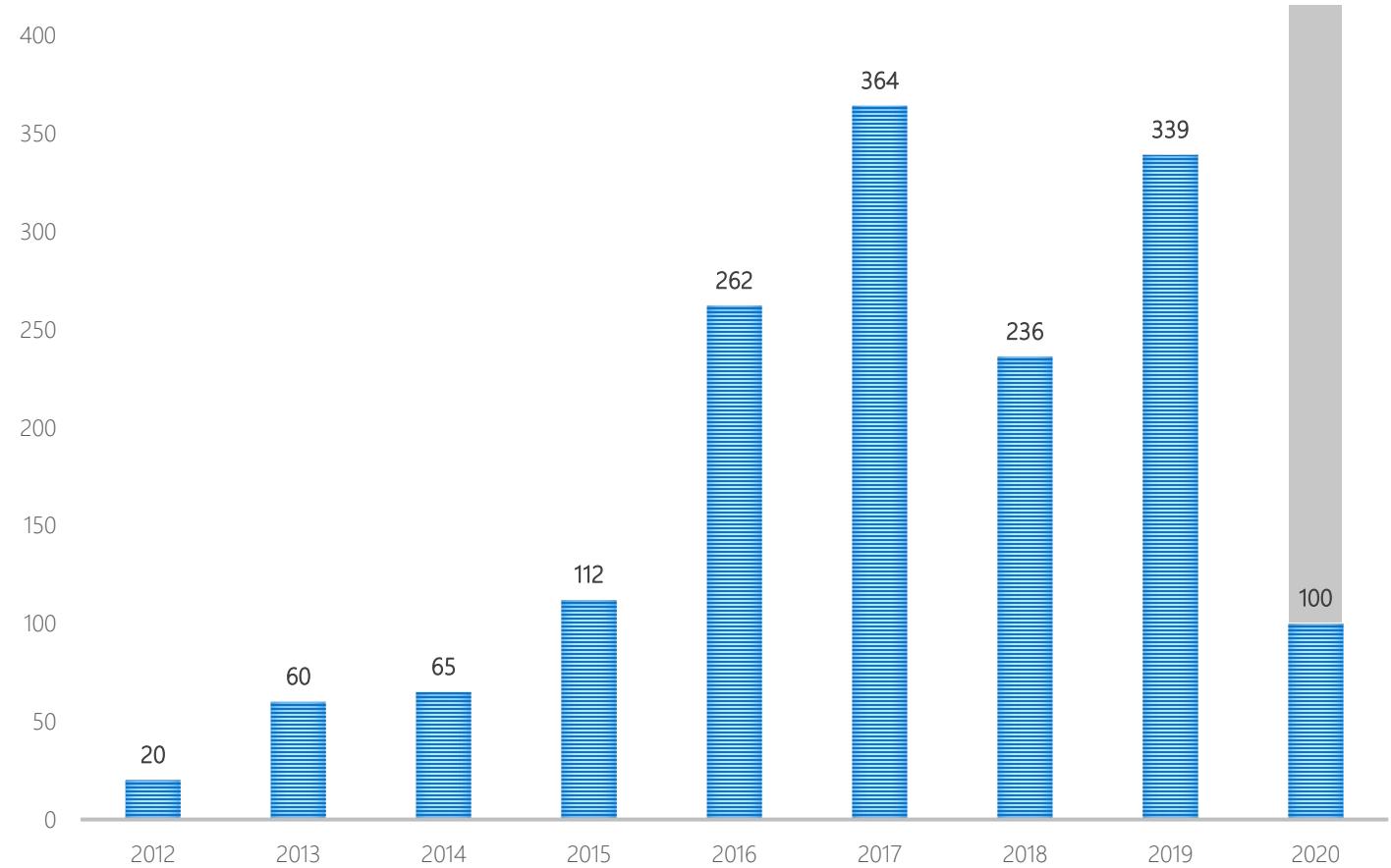
User satisfaction determines success

Features shipped every sprint

How do we know it's working?

# Features Delivered per Year

- More features in the 2016 calendar year than the previous four years combined



# Business Value of DevOps

- Embrace the new normal
- Build the culture you want
- Increase deployment with higher quality and security
- Break siloed development
- Introduce autonomy, mastery, and purpose
- Bring telemetry data early
- Get good at validated learning

# Knowledge Check

## Question #1: What is DevOps?

The union of people, process, and products to enable continuous delivery of value to your customers.

## Question #2: What are the main pillars of DevOps?

People, process, culture, tools

## Question #3: What are some of the practices of DevOps?

Shift-Left, Shift-Right, Continuous Integration, Continuous Delivery

