DevOps at Microsoft

09/09/2018 • 4 minutes to read

In this article

By: Sam Guckenheimer

This center will keep you current on how we continue to adopt DevOps at Microsoft. We've selected the best videos and articles from both public conferences and internal training sessions. For most of the videos, we also provide an accompanying article. We focus on practices that we use in Azure DevOps, which is the backbone of our One Engineering System (1ES), lessons learned from across Microsoft, and the best of our learning from Microsoft Research.

So that you don't get confused, I'll point out that Azure DevOps is the successor to Visual Studio Team Services (VSTS) and many of the articles refer to VSTS. I have not tried to monkey with the names just to achieve political correctness. For product news on Azure DevOps, take a look at our DevOps blog.

Why We Do DevOps at Microsoft

Martin Woodward leads a whistle-stop tour of Microsoft's seven-year DevOps journey, explaining why the company embarked on this transformation and what benefits it has already seen.

How We Work with Azure DevOps

The first block of talks drills deeper into our culture changes over the last eight years as we moved to Azure DevOps from a slower-moving world of on-premises software.

Article	Description
Moving to Cloud Cadence	Lori Lamkin covers our seven-year journey to cloud cadence. She goes through the sequence of steps we took, and that you can take as you become proficient at DevOps.
Agile principles in practice	Aaron Bjork discusses how we incorporate Agile principles and what that looks like in practice. Everything about how we manage teams, roles, planning, sprints, and flow has brought improvement to the software we build and use daily that customers can depend on.
Release flow: Our branching strategy	Working in a single master and using the pull request flow have been a key ingredient to keeping debt out and deployments clean. Ed Thomson covers lightweight topic branching on Git combined with Release Flow as part of our move to safe deployment. He starts with the transition from a hierarchical version control to Git.
Mindset shift to DevSecOps culture	Security is a key part of DevOps. Buck Hodges first walks through how we have done our security war games with red teams and blue teams. Buck goes on to cover our best practices for DevSecOps in running a SaaS business.
Live-Site Culture	Live-Site Culture (or a Production-First Mindset) is essential to running a service. Tom Moore discusses both how we handle service reliability and how we practice, You Build It, You Run It. You can't control what you can't measure. Along the way, Tom Moore shows how we use telemetry to monitor VSTS and gain continual insight into both the health and usage of the service.
Evolving test practices: combining development and test	Microsoft's decision to move to *a single engineering organization*, where development and testing are a unified part of the build process rather than separate roles, has helped every engineer have a greater impact on the quality of the software. Munil Shah shares his first hand experiences.

How We Architect Azure DevOps

These talks cover the accompanying technology behind Azure DevOps, its evolution from VSTS and from its on-premise origins.

Article	Description
From monolith to cloud service	Buck Hodges starts with the path from monolith to cloud service as we moved from a single delivery stream of TFS on-prem to dual streams including VSTS on Azure. He discusses how we maintain consistency between the on-prem product and the hosted multi-tenant service.
Achieving no downtime through versioned service updates	Running the hosted service 24x7x365 globally requires that we can deploy updates intraday with no downtime. Buck describes the architecture and technical process for updating the service while live.
Progressive experimentation with feature flags	A key advantage of the cloud service is that it provides a continuous feedback loop with our users. Here Buck discusses how we use feature flags to progressively reveal new functionality and to experiment in production.
Patterns for Resiliency in the Cloud	By definition, a 24x7x365 service needs to be always available. Buck describes how we have used cloud patterns for resiliency, such as circuit breakers and throttling, to ensure the availability and performance of VSTS.
Shift left to test fast and reliably	Availability would be meaningless without suitable quality. Munil Shah covers how we shift left to test fast and reliably - he takes us through the evolution that has led to our

Article	Description
	ability to run 60,000 tests in the pull request flow before commit to master and continuous integration.
Eliminating flaky tests to build trust	To get a reliable signal from high-volume test automation, we need to be able to trust the test results. Munil describes how we eliminate flaky tests so that red can mean red.
Shift right to test in production	Testing only in pre-production environments helps you with the faults you've previously encountered, but not the ones you haven't seen. Munil explains how there is no place like production and that means we need to test in production too.
Safe deployment practices	When you deploy continuously, you need to control the blast radius and continually expand based on the health of the release. Ed Glas goes over the safe deployment practices that we use for progressive exposure.

Researching Software Practices

We like to be informed by data. Measuring how well practices work is key. We're sharing our research and learning, so that you can be informed too.

Sam Guckenheimer works on Microsoft Visual Studio Cloud Services, including VS Team Services and Team Foundation Server. He acts as the chief customer advocate, responsible for strategy of the next releases of these products, focusing on DevOps. He has written four books on DevOps and Agile Software practices.