Continuous Delivery Using Azure DevOps Services

CDADS | 2 Days

This two-day course provides students with the DevOps principles and related hands-on practices to work better as a team, scale their agility, share and integrate their work, and deliver working software continuously in order to enable faster delivery of value and receive early and valuable feedback.

Course Objectives

By the end of the course, students will have had exposure-to and hands-on experience with the following …

- ✓ DevOps principles, benefits, and related practices
- ✓ The Three Ways: flow, feedback, and learning
- ✓ Technical excellence at scale
- ✓ Working as a team in an Azure DevOps environment
- ✓ Increasing flow through a technical value stream
- ✓ Practices and mindset of a Professional Scrum Team
- ✓ Scaled Professional Scrum (Nexus) framework
- ✓ Identifying and eliminating dependencies
- ✓ Minimizing dependencies through feature teams
- ✓ Configuring Azure DevOps for product development
- ✓ Planning work at scale: epics, features, user stories
- ✓ Product Backlog Refinement benefits and practices
- ✓ Creating a definition of "ready"
- ✓ Types of dependencies and the related risks
- ✓ Cross-team refinement to mitigate dependencies
- ✓ Working in small batches and limiting WIP
- ✓ Using Azure Boards to plan and execute a Sprint
- ✓ Creating and abiding-by a Definition of Done
- ✓ Reporting data via queries, charts, and widgets
- ✓ Using the Analytics service and related widgets
- ✓ Using dashboards as information radiators
- ✓ Collaborating as a cross-functional team
- ✓ The collective ownership mindset
- ✓ Git version control primer and workflows
- ✓ Branching strategies and potential side effects
- ✓ Visualizing and managing code dependencies
- ✓ Adopting an internal open source model
- ✓ Creating fast feedback loops
- ✓ Automated testing in Visual Studio and Azure Pipelines
- ✓ Writing and running unit tests in Visual Studio
- ✓ Creating and managing Azure Pipeline builds
- ✓ Creating and customizing YAML-based pipelines
- ✓ Queuing and monitoring automated builds
- ✓ Running automated tests as part of a build
- ✓ Collecting code coverage information
- ✓ Using Test Impact Analysis to improve performance
- ✓ Using test filters to run specific tests in a pipeline
- ✓ Practicing Continuous Integration (CI) and CI+
- ✓ Using Azure Pipelines for deployment
- ✓ Infrastructure as Code (IaC)
- ✓ Using Microsoft Azure for DevOps
- ✓ Automated deployment to Azure App Services
- ✓ Creating and deploying a release
- ✓ Continuous Delivery (CD) and using triggers
- ✓ Hypothesis-Driven Development (HDD)
- ✓ Customizing Azure DevOps to implement HDD
- ✓ Using feature flags to manage feature availability
- ✓ Telemetry and application performance management
- ✓ Using feature flags to support A/B testing
- ✓ Using LaunchDarkly and Application Insights for HDD
- ✓ Exploratory testing and taking testing "tours"
- ✓ Using the Microsoft Test and Feedback extension
- ✓ Understanding, identifying, and avoiding technical debt
- ✓ Using SonarCloud to gauge technical debt
- ✓ Patterns of effective collaboration
- ✓ Building a culture of learning and improvement
- ✓ Blameless retrospectives
- ✓ Building feedback directly into the product
- ✓ Tools for improvement, Communities of Practice (COPs)
- ✓ Tracking improvement through agile metrics

Case Study

The course contains *over 50 hands-on activities* where students work as a team, a pair, or individually using the tools that reinforce the DevOps principles and practices they are learning. Students will self-organize into cross-functional, collocated teams and work collaboratively on a case study using a shared Azure DevOps environment. All code will be provided.

Who Should Attend

This course is intended for experienced software development professionals who want to learn about DevOps in order to achieve Continuous Integration, Continuous Delivery, Continuous Feedback, and Continuous Learning in a technical value stream as supported by Azure DevOps Services, Visual Studio, and Azure in order to continually deliver working software at scale. Students will also install and utilize several extensions from the Azure DevOps Marketplace. Teams using Azure DevOps Server (Team Foundation Server) will also benefit from this course. Attendees should be familiar with C#, Visual Studio, Scrum, and have basic experience with Azure DevOps Services or Azure DevOps Server.

Accentient™

Continuous Delivery Using Azure DevOps Services

CDADS | 2 Days

Modules

Module 1: Increasing Flow at Scale

- ✓ The complexity of software development
- ✓ The need for empirical process control
- ✓ Increasing flow through a technical value stream
- ✓ Professional Scrum
- ✓ The Nexus scaled Scrum framework
- ✓ Practices for organizing teams
- ✓ Establishing feature teams to minimize dependencies

Module 2: Planning and Executing at Scale

- ✓ Organizing and refining the Product Backlog
- ✓ Creating a definition of "ready"
- ✓ Dependencies, types, and related risks
- ✓ Cross-team refinement to identify dependencies
- ✓ Planning and executing a Sprint
- ✓ Limiting work in progress (WIP)
- ✓ Working in small batches
- ✓ Creating and obeying a Definition of Done
- ✓ Using queries, charts, and dashboards for reporting

Module 3: Sharing Code

- ✓ Working collaboratively as a team
- ✓ Collective ownership mindset
- ✓ Git version control workflow (optional)
- ✓ Branching strategies and related side effects
- ✓ Using Code Maps to visualize code dependencies
- ✓ Using Package Management to share binaries
- ✓ Practicing internal open source (inner source)

Module 4: Integrating Continuously

- ✓ Why and how to create fast feedback loops
- ✓ The importance of automated testing
- ✓ Unit testing in Visual Studio
- ✓ Automated builds in Azure Pipelines
- ✓ Creating and customizing YAML-based builds
- ✓ Infrastructure as Code (IaC)
- ✓ Running tests during an automated build
- ✓ Code coverage and regression testing
- ✓ Configuring and using Test Impact Analysis
- ✓ Practicing Continuous Integration (CI) and CI+

Module 5: Delivering Continuously

- ✓ Azure Pipelines deployment
- ✓ Release definitions, stages, and releases
- ✓ Deployment targets, IaaS, PaaS, containers
- ✓ Using Microsoft Azure for DevOps
- ✓ Configuring service connections
- ✓ Automated deployment to an Azure App Service
- ✓ Release jobs, steps, and tasks
- ✓ Creating and deploying a release
- ✓ Release and stage triggers
- ✓ Practicing Continuous Delivery (CD)

Module 6: Empowering the Product Owner

- ✓ Build-Measure-Learn explained
- ✓ Hypothesis-Driven Development (HDD)
- ✓ Customizing Azure DevOps to implement HDD
- ✓ Feature flags overview
- ✓ Using LaunchDarkly to manage feature flags
- ✓ Telemetry and application performance management
- ✓ Using Application Insights to gather telemetry
- ✓ A/B testing explained
- ✓ Using feature flags to support A/B testing
- ✓ Exploratory testing and taking testing "tours"
- ✓ Using the Microsoft Test and Feedback extension
- ✓ Understanding and identifying technical debt
- ✓ Using SonarCloud to gauge your technical debt
- ✓ Making technical debt transparent
- ✓ Practices for paying off technical debt

Module 7: Learning and Improving Continuously

- ✓ Working and learning as a team
- ✓ Patterns of effective collaboration
- ✓ Pairing, swarming, and mobbing practices
- ✓ Building a culture of learning and improvement
- ✓ Blameless retrospectives
- ✓ Building feedback directly into the product
- ✓ Communities of Practice (COPs)
- ✓ Tracking improvement through agile metrics
- ✓ Using the wiki to build tribal knowledge

Course Designer

This course was designed by Richard Hundhausen, Microsoft's first Visual Studio ALM/DevOps MVP, Professional Scrum Trainer, co-creator of the Nexus Scaled Professional Scrum framework, and an experienced software developer. To see other developer courses, visit www.accentient.com.

Accentient™