

Course duration

- 2 days

Course Benefits

- Learn to increase flow at scale.
- Learn to plan and execute at scale.
- Learn to share code.
- Learn to integrate continuously.
- Learn to deliver continuously.
- Learn to empower the product owner.

Course Outline

1. Increasing Flow at Scale
 1. The complexity of software development
 2. The need for empirical process control
 3. Increasing flow through a technical value stream
 4. Professional Scrum
 5. The Nexus scaled Scrum framework
 6. Practices for organizing teams
 7. Establishing feature teams to minimize dependencies
2. Planning and Executing at Scale
 1. Organizing and refining the Product Backlog
 2. Creating a definition of “ready”
 3. Dependencies, types, and related risks
 4. Cross-team refinement to identify dependencies
 5. Planning and executing a Sprint
 6. Limiting work in progress (WIP)
 7. Working in small batches
 8. Creating and obeying a Definition of Done
 9. Using queries, charts, and dashboards for reporting
3. Sharing Code
 1. Working collaboratively as a team
 2. Collective ownership mindset?Git version control workflow (optional)
 3. Branching strategies and related side effects
 4. Using Code Maps to visualize code dependencies?Using Package Management to share binaries
 5. Practicing internal open source (inner source)
4. Integrating Continuously
 1. Why and how to create fast feedback loops

2. The importance of automated testing
3. Unit testing in Visual Studio
4. Automated builds in Azure Pipelines
5. Creating and customizing YAML-based builds
6. Infrastructure as Code (IaC)
7. Running tests during an automated build
8. Code coverage and regression testing
9. Configuring and using Test Impact Analysis
10. Practicing Continuous Integration (CI) and CI+
5. Delivering Continuously
 1. Azure Pipelines deployment
 2. Release definitions, stages, and releases
 3. Deployment targets, IaaS, PaaS, containers
 4. Using Microsoft Azure for DevOps
 5. Configuring service connections
 6. Automated deployment to an Azure App Service
 7. Release jobs, steps, and tasks
 8. Creating and deploying a release
 9. Release and stage triggers
 10. Practicing Continuous Delivery (CD)
6. Empowering the Product Owner
 1. Build-Measure-Learn explained
 2. Hypothesis-Driven Development (HDD)
 3. Customizing Azure DevOps to implement HDD
 4. Feature flags overview
 5. Using LaunchDarkly to manage feature flags
 6. Telemetry and application performance management
 7. Using Application Insights to gather telemetry
 8. A/B testing explained
 9. Using feature flags to support A/B testing
 10. Exploratory testing and taking testing “tours”
 11. Using the Microsoft Test and Feedback extension
 12. Understanding and identifying technical debt
 13. Using SonarCloud to gauge your technical debt
 14. Making technical debt transparent
 15. Practices for paying off technical debt
7. Learning and Improving Continuously
 1. Working and learning as a team
 2. Patterns of effective collaboration
 3. Pairing, swarming, and mobbing practices
 4. Building a culture of learning and improvement
 5. Blameless retrospectives
 6. Building feedback directly into the product
 7. Communities of Practice (COPs)
 8. Tracking improvement through agile metrics
 9. Using the wiki to build tribal knowledge

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Azure DevOps Services class:

- Team-based development experience.
- Familiarity with Visual Studio and Scrum and have basic experience with Azure DevOps Services, Visual Studio Team Services, or Team Foundation Server.