

Course Outline:

1: What is Continuous Delivery?

This module will review the definition of Continuous Delivery, and why it is an essential part of high-quality software design. For teams that incorporate an Agile programming approach to software development, Continuous Delivery is an integral part of the process.

1. Introduction to Continuous Delivery
2. Where does Continuous Delivery fit in the DevOps landscape?
3. How does Continuous Delivery work?
4. What are the benefits of using Continuous Delivery?
5. Anti-patterns
6. Explore CI Settings in Team Services
7. Explore Gated Check-in Settings in Team Services

Lab: Explore a Sample CI Configuration

2: Agile Development

This Module is a high-level overview of the Agile development methodology and how it fits into Continuous Delivery and the DevOps landscape.

1. Review: Agile development and the Agile Manifesto
2. Integrated development environments
3. Source control and versioning
4. Test-driven development
5. Pair programming
6. User stories
7. Developer tasks
8. Sprints
9. Create a Team Services Account
10. Add User Stories to Team Services Project
11. Add Developer Task to User Stories
12. Test Driven Development

Lab: Be Agile with Team Services

3: Continuous Integration

This Module focuses on the definition of Continuous Integration, its relationship to Continuous Delivery, and why it is so important.

1. What is Continuous Integration?
2. How Continuous Integration fits in the DevOps landscape
3. How does Continuous Integration work?
4. Continuous Integration essentials
5. Common Continuous Integration practices
6. Benefits of Continuous Integration
7. From Continuous Integration to Continuous Delivery
8. Configure Check-in Policy
9. Configure Build Trigger
10. Configure Inspections and Test Execution

Lab: Configure Team Services for Continuous Integration

4: Configuration and Implementation

This Module discusses the prerequisites of Continuous Delivery, and how mastering the process translates into higher quality and lower cost.

1. Version Control
2. Automated Builds
3. Automated Testing
4. Automated Acceptance Testing
5. Package Repository
6. Managing Dependencies
7. Managing Environments
8. Configuration Principles
9. Configure a Package Repository using Team Services
10. Invoke a build that deploys to the Package repository

Lab: Configure a Package Repository

5: Deployment Pipeline and Scripting

In this Module, we will take a deep dive into the implementation, best practices, and environment of a Deployment Pipeline and Scripting Deployment.

1. What is a Deployment Pipeline?
2. Committing Code
3. Gated Acceptance Testing
4. Automating Deployment
5. Testing Stages
6. Implementing a Deployment Pipeline
7. Build Tools Overview
8. Deployment Scripting
9. Build Scripting
10. Automating Tests
11. Configure Build Tools (MSBuild)
12. Create Deployment Script
13. Deploy Sample Application

Lab: Automating Deployment

6: Automated Testing

In this Module, we will review necessary components of acceptance testing that affect Developers, Project Managers, and Testers

1. Creating Acceptance Tests
2. Automating Acceptance Tests
3. Creating Unit Tests
4. Automating Unit Tests
5. Automating Capacity Testing
6. Parallel Testing
7. Refactoring

8. Automating Unit Tests
9. Automating Acceptance Tests

Lab: Automating Test Execution

7: Delivery vs. Deployment

This Module covers the differences between Continuous Delivery and Continuous Deployment, and the relationship that exists between them.

1. Releasing an Application
2. Deploying an Application
3. Continuous Delivery
4. Continuous Deployment
5. Virtual Environments
6. Roll back a Deployment
7. Deliver Deployment Packages to the Package Repository
8. Deploy the current version from the Package Repository
9. Deploy a previous version from the Package Repository

Lab: Delivery vs Deployment

8: Effective Continuous Delivery Environments

In this Module, students will learn how to apply Continuous Delivery to their current environment, and the maintenance that will be necessary to sustain it.

1. Infrastructure Management
2. Communicating with the Operations Team
3. Configuration Management
4. Infrastructure in the Cloud
5. Infrastructure Maintenance
6. Configure Azure Virtual Server
7. Save Virtual Server Image

Lab: Cloud Services

9: Managing Implementation

This module deals with components, dependencies, and versioning issues in Continuous Deployment.

1. Managing Components
2. Managing Dependencies
3. Version Control
4. Version Control Options
5. Mainline Development
6. Merging and Branching

10: Managing Continuous Delivery

This module demonstrates the importance of maintenance & how it affects each release. It is important that project managers, developers, testers, and clients have a system which enables them to stay communicative and on the same page.

1. Risk Management
2. Delivery lifecycle
3. Common Pitfalls
4. Documentation is crucial!
5. Maintaining your Configuration

Lab Outline:

- 1: Explore a Sample CI Configuration
- 2: Be Agile with Team Services
- 3: Configure Team Services for Continuous Integration
- 4: Configure a Package Repository
- 5: Automating Deployment
- 6: Automating Test Execution
- 7: Delivery vs Deployment
- 8: Cloud Services

