



Team Foundation Server 2015 Application Lifecycle Management (ALM)

Team Foundation Server (TFS) is more than just version control! This course provides an overview of the Application Lifecycle Management (ALM) tooling that's available using Team Foundation Server 2015. We'll start with version control and move on to cover TFS builds, project planning and tracking using TFS Work Items, QA testing and Virtual Lab Management using Microsoft Test Manager, Storyboarding with TFS & PowerPoint, and finally gathering and tracking customer feedback using Feedback Manager. Along the way, we'll also discuss the features of TFS Source Control (including branching & merging), and using Visual Studio & TFS to enforce code quality and manage code reviews.

What you'll learn:

- Microsoft Application Lifecycle Management (ALM) Overview
 - What is ALM and how can TFS help?
 - Team Foundation Server
 - Version Control
 - Requirements Management
 - Agile & Scrum Planning
 - Automated Builds
 - Testing & Defect Tracking
 - Feedback Management
 - Lab Management
 - Reporting
- Setup & Configuration
 - Core pieces of TFS
 - Things that bolt on to TFS
 - TFS on Windows Azure
 - Configuration types
 - Hardware & Pre-requisites
 - Installing TFS
 - Team Projects & Team Project Collections
 - Connect to TFS using Team Explorer
 - Cross-platform TFS
 - Process Templates
 - Permissions & Security

- Backup & Restore
- Version Control
 - TFS version control
 - TFS with Git
 - Centralized version control vs. Distributed Version Control (DVCS)
 - My first 'Get Latest'
 - TFS Workspaces
 - Server Workspaces vs. Local Workspaces
 - Check-in, Check-out
 - Repository setup
 - Offline features
 - Version control locks
 - Shelving
 - Branching & Merging
 - Server-side settings
 - Permissions
 - Version Control from Eclipse
 - Version Control from the command line (tf.exe)
- Work Items
 - What is a Work Item?
 - Work Item Types
 - Work Item Links
 - Areas, Iterations, and Teams
 - Work Item Queries
 - Team Explorer
- Coding, Code Reviews, and Metrics
 - My Work
 - Suspend and Resume
 - Code Reviews
 - Code Clone Analysis
 - Code Analysis
 - Code Metrics
- Scrum and Agile Planning Tools
 - What is Scrum?
 - Teams, Work Items, and Scrum
 - Product Backlog Manager
 - Sprint Planning
 - Scrum Board
 - The Burndown Chart
 - Kanban Board
 - Team capacity planning

- Tracking Velocity, Story Points, and Estimation
- Requirements Management & Feedback
 - PowerPoint Storyboarding
 - SharePoint
 - Link a Storyboard to a Requirement
 - Feedback Manager
 - Sending, Receiving, and Tracking Stakeholder Feedback
- Work Item & Process Template Customization
 - Structure of a Process Template
 - Structure of a Work Item
 - What can you customize?
 - WitAdmin.exe
 - How to customize?
 - States & State Transitions
 - Work Item Form customization
 - Work Item Permission customization
 - Remove auto-resolve on check-in
- Automated Builds
 - Why automated builds?
 - Installing and Planning TFS Build
 - Create and edit a build definition
 - Run the build
 - XAML builds vs. the new Powershell-based Builds
 - Gated Check-in Builds
 - Build Customization
 - Clone a build
 - Use a Build Editor project
 - Use 3rd Party Build Activities
 - Add an argument to a build script
 - Create a custom Build Activity

Details

Technologies: Team Foundation Server 2015, Visual Studio 2015

Programming language: n/a

Duration: 2 days, 9am to 5pm

Lab Machine Requirements

Operating System: Windows 8, Windows Server 2015

Memory: 8GB or more

Required applications: Visual Studio Ultimate 2015 with latest service packs applied

Other: Lab machines should have the Hyper-V role installed and be up to date with patches from Windows Update. Students should have administrator rights on their laptop.