APPLICATION LIFECYCLE MANAGEMENT USING VISUAL STUDIO 2010 (AGILE)

ALM2010A | 3 Days

INTRODUCTION

This three-day, instructor-led course provides students with the knowledge and skills to effectively use the Application Lifecycle Management (ALM) tools found in Visual Studio 2010. The course focuses on teaching project managers, architects, .NET developers, SQL Server developers, testers and release managers the various ALM features and capabilities of Visual Studio 2010 and Team Foundation Server 2010.

AUDIENCE

This course is intended for current software development professionals, including project managers, architects, database professionals, software developers, testers and release managers who are involved in building Windows or web applications. Regardless of the student's role, he or she will be able to learn and get hands-on experience with all of the role-based features of Visual Studio 2010.

AT COURSE COMPLETION

After attending this course, students will be able to:

- Understand Visual Studio 2010's Application Lifecycle Management capabilities
- Understand the architecture of Team Foundation Server
- Use the Team Foundation Server Administration Console
- Plan and create team project collections and team projects
- Choose an appropriate process template
- Manage and secure a team project
- Create and query work items using various client applications (Excel, Web Access, etc.)
- Understand and use work item hierarchies for maximum traceability
- Plan and track an Agile project using Excel workbooks
- Use UML to model use requirements and describe a system
- Use layer diagrams to visualize and define a system's logical architecture
- Use Architecture Explorer and sequence diagrams to visualize code and dependencies
- Use database projects to manage SQL Server database schema changes
- Use automation to build, deploy, generate test data and unit test SQL Server databases
- Understand the architecture and usage of Team Foundation Version Control
- Work with version control from Visual Studio 2010 in a multi-user environment
- Use .NET unit tests, code analysis and code metrics to ensure high-quality code
- Use code coverage and Test Impact Analysis to ensure high-quality tests
- Use IntelliTrace to reduce the time required to reproduce and diagnosis errors
- Improve code performance using Performance Profiler and load testing
- Plan your testing effort using test Plans, Suites, Cases and Configurations
- Use Microsoft Test Manager to plan, create and execute your manual tests
- Use coded UI tests to automate UI testing
- Test web applications using automated tools
- Configure build controllers and agents
- Automate the software build process using Team Foundation Build
- Configure gated check-ins and private builds to alleviate build breaks

PREREQUISITES

Before attending this course, the student should have experience working on a team-based software development project and be familiar with their organization's Application Lifecycle Management process and tools. Additionally, students should:

- Have played one or more roles in the software development lifecycle
- Have familiarity with one or more software development methodologies
- Have familiarity with distributed application design (i.e. client/server, web, n-tier, etc.)
- Have used Visual Studio
- Be comfortable reading and understanding business requirements
- Can read and understand C# .NET code (all source code will be provided)
- Understand Microsoft Windows operating system and security basics

COURSE OUTLINE

Module 1: Introduction

This module introduces Visual Studio 2010 and Team Foundation Server 2010 from a business and technical point of view, including the various editions and components that can be installed and configured.

Lessons

- The challenges teams face
- What is Application Lifecycle Management
- Visual Studio's support for ALM
- Architectural overview of Team Foundation Server
- Features and capabilities by edition and team role

Lab Exercises

- Connect to a team project using Team Explorer
- Manage documents
- Create and execute queries
- Execute reports
- Customize the project portal

Module 2: Team Projects

This module introduces team projects, why they are used and the tasks that project administrators perform. Students will learn how to plan, create and configure team projects using Team Explorer.

<u>Lessons</u>

- Project administrator role
- Team project collections
- Team projects
- Configuring team projects
- Managing team projects

<u>Lab Exercises</u>

- Manage security and permissions
- Create a team project collection
- Create a team project
- Explore the process guidance
- Setup classification areas and iterations

Module 3: Process Templates and Work Items

This module introduces students to the various software development methodologies and the level of support found in Team Foundation Server. Students will learn the differences between the Agile, CMMI and Scrum process templates. Students will also learn to work with work items and work item hierarchies.

<u>Lessons</u>

- Software development methodologies
- Process templates
- Agile vs. formal software development
- MSF for Agile Software Development v5.0
- MSF for CMMI Process Improvement v5.0
- Implementing Scrum
- Work item types and hierarchies
- Creating and managing work items

<u>Lab Exercises</u>

- Create and manage work items using Team Explorer
- Create and manage work items using Team Web Access
- Create and manage work items using Work Item View Only
- Create and manage work items using Microsoft Excel
- Create and manage work items using Microsoft Project
- Create and manage work items using Microsoft Test Manager

Module 4: Version Control

This module introduces the benefits of revision control as well as how to use Team Foundation Version Control. Students will learn how to setup workspaces and perform get, check-out and check-in operations from within Visual Studio 2010. Students will also learn advanced features such as managing and resolving conflicts, shelving, branching and merging.

Lessons

- Team Foundation Version Control
- Integration with Visual Studio
- Workspaces
- Source Control Explorer
- Basic operations: get, check-out, check-in, label
- Managing and resolving conflicts
- Shelving, branching and merging

Lab Exercises

- Create a workspace
- Place a solution under source control
- Get, check-out and check-in files
- View history and compare files
- Detect and resolve conflicts
- Shelve and unshelve changes

Module 5: Modeling the Application

This module introduces the various architecture tools found in the Ultimate edition. Students will learn how to explore existing (Brownfield) applications by generating graphs and diagrams and by using Architecture Explorer. Students will also learn how to use UML models to help the team understand, clarify and communicate ideas about the code and the user requirements. Layer diagrams will also be introduced as a mechanism to not only define the logical architecture, but to also validate any software development against them.

Lessons

- Architect role and responsibilities
- Directed Graph Modeling Language (DGML)
- Exploring existing code
- Architecture Explorer
- Sequence diagrams
- UML models and diagrams
- Layer diagrams

Lab Exercises

- Generate and share DGML documents
- Use Architecture Explorer
- Generate a sequence diagram
- Create a UML model
- Create UML diagrams
- Create a layer diagram
- Validate development against a layer diagram

Module 6: Agile Database Development

This module discusses the database developer role and the capabilities of the Premium edition to support the related activities. Students will learn how to use database projects to manage their database development including refactoring, building, deploying, managing schema and data changes, loading test data and running unit tests.

Lessons

- The database development life cycle
- Database projects
- Importing schemas and scripts
- Database refactoring
- Data generation plans
- Database unit testing
- Building and deploying
- Comparing schemas and data

Lab Exercises

- Create a database project
- Import an existing database schema
- Refactor database objects
- Build and deploy the database project
- Generate test data
- Create and run stored procedure unit tests

Module 7: Writing Quality Code

This module introduces the developer to various tools found in Visual Studio 2010 that will help improve code quality. Students will learn how to create proper unit tests, scan their code for common defects and best practice violations, calculate code metrics to reveal complex areas in their code and profile their applications for performance problems. Students will also learn how to use Test Impact Analysis and code coverage to ensure the quality of their test runs and how to use IntelliTrace to capture a history of execution data for enhanced debugging.

Lessons

- The developer and tester roles and responsibilities
- Unit testing, code coverage and Test Driven Development (TDD)
- Test Impact Analysis
- Code analysis and code metrics
- Application profiling
- IntelliTrace

Lab Exercises

- Create unit tests, refactor and re-run tests
- Create a data-driven unit test
- Configure code coverage
- Use Test Impact Analysis
- Run code analysis and code metrics
- Profile a poorly performing .NET application
- Collect IntelliTrace diagnostic information
- Debug using historic IntelliTrace data

Module 8: Testing the Application

This module introduces the various testing tools you can use to help ensure that applications meet requirements and are free of defects. Students will learn the lifecycle of test planning, execution and tracking progress, as well as how to record, configure and run manual tests, web performance tests and load tests.

Lessons

- The testing lifecycle
- Manual tests
- Coded UI tests
- Web performance tests
- Load tests

Lab Exercises

- Create a test plan and a manual test case
- Execute the manual test
- Automate the manual test
- Record and execute a Web performance test
- Place a Web performance test under load

Module 9: Team Foundation Build

This module introduces the architecture and usage of Team Foundation Build. Students will learn how to define builds, queue them manually or based on a trigger and analyze the finished builds, setting the quality appropriately.

Lessons

- Team Foundation Build
- The Build Process
- Build definitions and templates
- View, manage and queue builds
- Build reports
- Automating Team Foundation Build
- Continuous Integration (CI)

Lab Exercises

- Define a workflow-based build
- Define a (legacy) MSBuild-based build
- Queue a build in various ways
- Set the quality of a completed build
- Schedule a build to queue at a specific time
- Enable Continuous Integration

Course Designer

This course was designed by Richard Hundhausen of Accentient, Inc. Richard is a Visual Studio ALM MVP and Microsoft Regional Director, as well as an experienced developer and trainer.

For more information, contact Mark Dunn at 770 653-6364 or email mark@dunntraining.com.