# VISUAL STUDIO TEAM SYSTEM 2008 END-TO-END USING MSF/AGILE


Course E2EA08: Three days; Instructor-Led Course Syllabus

## INTRODUCTION

This three-day, instructor-led course provides students with the knowledge and skills to effectively use Visual Studio 2008 Team System to manage their entire Software Development Life Cycle (SDLC) in a team-based environment. The course focuses on teaching project managers, software architects, database professionals, software developers, testers, and release managers the various features and capabilities of Team System and Team Foundation Server.

## AUDIENCE

This course is intended for current software development professionals, including project managers, architects, database administrators, database developers, software developers, testers, and release managers who are involved in building Windows or web-based 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 2008 Team System.

## AT COURSE COMPLETION

After attending this course, students will be able to:

- Understand Team System's capabilities
- Understand Team System's support for the Software Development Life Cycle
- Understand the architecture of Team Foundation Server
- Plan and create a team project
- Manage and secure a team project
- Choose an appropriate software development methodology and process template
- Create and query work items using various client applications
- Use the Distributed System Designers to architect a connected .NET solution
- Validate solution deployment and generate a deployment report
- Use database projects to manage change to SQL Server database schemas
- Use automation to build, deploy, generate test data, and unit test SQL Server databases
- Leverage Visual Studio 2008's new software development features
- Understand the architecture and usage of Team Foundation Version Control
- Work with version control from Visual Studio 2008 in a multi-user environment
- Improve code quality through unit testing, code analysis, code metrics, and profiling
- Test web applications using automated tools
- Place web and unit tests under load to stress your architecture and plan for capacity
- Automate the software build process using Team Foundation Build

## PREREQUISITES

Before attending this course, students should have experience working on a team-based software development project and be familiar with their organization's Software Development Life Cycle.

- Have played one or more roles in the SDLC: PM, architect, DBA, developer, tester
- Have familiarity with one or more methodologies. (example: MSF, XP, Scrum, RUP)
- Have familiarity with distributed application design. (example: client/server, web applications, web services, etc.)
- Have used Visual Studio
- Be comfortable reading user requirements and business-need documents.
- Understand the basic foundations of .NET
- Can read and understand C# .NET code (all source code will be provided)
- Understand Microsoft Windows operating system and security basics
- Have some experience with a reporting tool. For example: SQL RS, Access, Crystal

## COURSE OUTLINE

### Module 1: Introduction

This module introduces Visual Studio 2008 Team System from a business and technical point of view.

Lessons

- The challenges with building software
- What is Team System
- Architectural overview of Team System and Team Foundation Server
- Team System features by role

Lab Exercises

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

**Module 2: Team Projects**

This module introduces the Project Manager and Project Administrator roles and team projects. Students will learn how to plan, create and configure team projects using Team Explorer.

Lessons

- Project manager role
- Project administrator role
- Team projects
- Configuring team projects
- Managing team projects
- Client applications

Lab Exercises

- Manage Team Foundation Server level security
- Create a team project
- Explore and modify the process guidance
- Enable check-in policies
- Setup classification areas and iterations
- Secure the classification areas

**Module 3: Methodologies and Work Items**

This module introduces students to various popular software development methodologies, and the level of support in Team System. Students will learn about Microsoft Solutions Framework, process templates, and managing work items.

Lessons

- Software development methodologies
- Microsoft Solutions Framework (MSF)
- Methodology support in Team System
- Process templates
- Work items
- Creating and managing work items

Lab Exercises

- Create a scenario using Team Explorer
- Create a risk using Microsoft Excel
- Create a task using Microsoft Project
- Query work items using Team Explorer
- Query and manage work items using Web Access Power Tool

**Module 4: Architecture**

This module introduces the Architect role and the tools found in the Architecture edition. Students will learn how to use the various Distributed System Designers to create and validate models of their systems, applications, and deployment environments.

Lessons

- Architect role and responsibilities
- Distributed system designers
- Logical datacenter designer
- System designer
- Application designer
- Trial deployment
- DSL vs. UML

Lab Exercises

- Create a logical datacenter diagram
- Create a system diagram
- Create an application diagram
- Perform a trial deployment
- Validate the deployment
- Generate a deployment report
- Implement a web application from a diagram

**Module 5: Database Edition**

This module discusses the database professional role, including administrators and developers, and the capabilities of the Database edition to support these roles. 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
- Comparing schemas and data
- Database unit testing
- Database refactoring
- Data generation plans
- Building and deploying

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 6: Version Control**

This module introduces all roles to the benefits and usage of Team Foundation Version Control. Students will learn how to setup workspaces and perform get, check-out, and check-in operations from within Visual Studio 2008. Students will also learn advanced features, such as branching, merging, conflict detection and resolution, and shelving.

Lessons

- Benefits and architecture of Team Foundation Version Control
- Comparison to VSS
- Using version control
- Integration with Visual Studio
- Get, check-out, check-in, label
- Branching, merging, conflicts, shelving

Lab Exercises

- Create a workspace
- Place a solution under source control
- Check out and check in files
- View history and compare files
- Resolve conflicts
- Shelve and unshelve changes

**Module 7: Development Edition**

This module introduces the Developer role and the tools found in the Development edition that will help improve code quality. Students will learn how to create proper unit tests that have adequate code coverage, scan their code for common defects and best practice violations, calculate code metrics to reveal areas which are overly complex, and profile their code for performance problems.

Lessons

- The developer and tester roles and responsibilities
- Developer features in Visual Studio 2008
- Unit testing, code coverage, and Test Driven Development (TDD)
- Code analysis
- Code metrics
- Application profiling

Lab Exercises

- Create a unit test
- Refactor the method and re-run the unit test
- Create a data-driven unit test
- Calculate code coverage when running a unit test
- Run code analysis on a .NET assembly
- Calculate code metrics
- Profile a poorly performing .NET application
- Using Team Foundation Version Control and code profiling tools

**Module 8: Test Edition**

This module introduces the Tester role and the tools found in the Test edition that will help verify applications meet requirements and are free of defects. Students will learn how to record, configure, and run HTTP-based web tests, bind web tests to data sources, place web and unit tests under load, and use manual and generic tests appropriately.

Lessons

- Web testing
- Data bound and coded web tests
- Load testing
- Manual tests
- Generic tests

Lab Exercises

- Create a web site based on the personal web site template
- Record a web test
- Create a data-driven web test
- Create a load test, placing the web test under load
- Modify load test properties
- Create and execute a manual test

**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

- Introduction to Team Foundation Build
- The Build Process
- Reporting
- Automating Team Foundation Build and Continuous Integration (CI)

Lab Exercises

- Create a build definition
- Queue a team build in various ways
- Set the quality of a completed build
- Schedule a build to queue at a specific time
- Configure build notifications
- Enable continuous integration

**Course Designer**

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

For more information, visit www.accentient.com