

## **Workshop 2631A:**

# **Optimizing the Software Development Lifecycle with Microsoft Visual Studio Team System**

Length:	3 Days
Published:	March 22, 2006
Language(s):	English
Audience(s):	Developers
Level:	200
Technology:	Microsoft Visual Studio 2005
Type:	Workshop
Delivery Method:	Instructor-led (classroom)

### **About This Workshop**

Elements of this syllabus are subject to change.

This three-day instructor led workshop provides students with the knowledge and skills to use Microsoft Visual Studio Team System tools to optimize software application development.

### **Audience Profile**

This workshop is intended for the five job roles that comprise a typical software application development team: project manager, infrastructure architect, solution architect, developer, and tester. Students will have one to five years of experience working in these job roles with organizations that have large user bases (5,000+).

### **At Workshop Completion**

After completing this course:

Students in each job role will be able to describe how Visual Studio Team System supports their job roles through the tracks of the software development life cycle.

Project managers will be able to plan an application development project using Visual Studio Team System tools.

Infrastructure and solution architects will be able to design an application using Visual Studio Team System tools.

Developers will be able to develop application code using Visual Studio Team System tools.

Testers will be able to stabilize application code using Visual Studio Team System tools.

Solution architects will be able to create an application deployment package using Visual Studio Team System tools.

## **Workshop Outline**

### **Session 1: Introduction to Visual Studio Team System**

This unit provides an overview of the software development lifecycle and the Visual Studio Team System tools that support each job role through the tracks of the software development lifecycle. The unit also details the workshop-wide scenario that provides the context for lab activities.

### **Lessons**

- Introduction to the Woodgrove Instant Securities Portal
- Overview of the WISP Development Lifecycle
- The Envisioning Track
- The Planning Track-Project Setup
- The Planning Track-Design
- The Building Track
- The Stabilizing Track
- The Deploying Track
- The WISP Application
- Visual Studio Team System Components
- Demonstration: Exploring Visual Studio Team System
- WISP Project Kickoff Discussion

### **Lab : Exploring the Resource Toolkit and GABBI**

- Exploring the Resource Toolkit
- Adding Blog Entries to GABBI

After completing this unit, students will be able to:

- Identify the features of WISP.
- Identify the tools that are included in Visual Studio Team System.
- Use the Resource Toolkit to perform the tasks in this workshop.

### **Session 2: Setting Up an Application Development Project**

This unit provides an overview of the planning track (project setup) of a software application development project. The unit also describes the key tasks that the project managers perform during the planning track by using Visual Studio Team System tools.

### **Lessons**

- Project Setup Tasks
- Process Methodologies Supported by Visual Studio Team System
- Demonstration: Setting Up a Team Project
- WISP Project Setup Review Discussion
- Best Practices for Using Visual Studio Team System Tools to Set Up an Application Development Project

### **Lab : Setting Up the WISP ProjectLab : Project Manager**

- Creating the WISP Project
- Defining Work Items and Creating a Document Library

### **Lab : Infrastructure Architect**

- Creating the GABBI Logical Datacenter Diagram

### **Lab : Solution Architect**

- Creating the GABBI Application Diagram

### **Lab : Developer and Tester**

- Running Unit and Code-Coverage Tests on GABBI

After completing this unit, students will be able to:

- Project managers will be able to:
- Create a team project.
- Define work items.
- Create a document library.
- Infrastructure architects will be able to use Visual Studio Team System tools to create a logical datacenter diagram.
- Solution architects will be able to use Visual Studio Team System tools to create an application diagram.
- Developers and testers will be able to use Visual Studio Team System tools to run unit and code coverage tests.

### **Session 3: Designing an Application**

This unit provides an overview of the planning track (design) of a software application development project. The unit also describes the key tasks that the infrastructure architects and solution architects perform during the planning track by using Visual Studio Team System tools.

#### **Lessons**

- Design Tasks
- Design Diagrams
- What Are Source Control Policies?
- Demonstration: Creating Design Diagrams
- WISP Design Review Discussion
- Best Practices for Using Visual Studio Team System Tools to Design an Application

### **Lab : Designing WISPLab : Project Manager**

- Defining Additional WISP Requirements and Tasks
- Creating WISP Project Documents by Using Process Methodology Templates

### **Lab : Infrastructure Architect**

- Creating the WISP Logical Datacenter Diagram
- Testing the WISP Application Deployment

### **Lab : Solution Architect**

- Creating the WISP Application Diagram
- Uploading the WISP Web Services Specifications and Defining Policies and Tasks

### **Lab : Developer**

- Running a Code Analysis Test on GABBI

### **Lab : Tester**

- Running a Load Test on GABBI

After completing this unit, students will be able to:

- Project managers will be able to use Visual Studio Team System tools to add requirements, tasks, and document templates to a Team Portfolio Project.
- Infrastructure architects will be able to:
  - Create a logical datacenter diagram.
  - Create a deployment diagram.
  - Use source control.
- Solution architects will be able to:
  - Create an application diagram.
  - Configure a check-in policy.
  - Use source control.
  - Upload documents to the project portal site.
  - Manage task assignments.
- Developers will be able to use Visual Studio Team System tools to run a static analysis test.
- Testers will be able to use Visual Studio Team System tools to run load tests.

### **Session 4: Building an Application**

This unit provides an overview of the building track of a software application development project. The unit also describes the key tasks that the developers perform during the building track by using Visual Studio Team System tools.

### **Lessons**

- Build Tasks
- What Is a Class Diagram?
- The Test-Driven Development Methodology
- Demonstration: Writing Source Code by Using the TDD Methodology
- WISP Build Review Discussion
- Best Practices for Using Visual Studio Team System Tools to Build an Application

### **Lab : Building WISPLab : Project Manager**

- Viewing Reports
- Exporting a Report

### **Lab : Infrastructure Architect**

- Extending the WISP Logical Datacenter Diagram
- Testing the Extended WISP Application Deployment

### **Lab : Solution Architect**

- Extending the WISP Application Diagram
- Testing the Extended WISP Application Deployment

### **Lab : Developer**

- Generating the WISP Application Framework
- Writing and Testing the WISP Source Code

### **Lab : Tester**

- Running a Load Test
- Viewing Reports
- Exporting a Report

After completing this unit, students will be able to:

- Project managers will be able to use the predefined reports included in Visual Studio Team System.

- Infrastructure architects will be able to use Visual Studio Team System tools to extend a logical datacenter diagram.
- Solution architects will be able to use Visual Studio Team System tools to extend an application connection diagram.
- Developers will be able to:
  - Generate an application framework.
  - Create a class diagram.
  - Run unit tests.
  - Run code-coverage tests.
  - Use source control.
  - Manage task assignments.
- Testers will be able to use the predefined reports included in Visual Studio Team System.

### **Session 5: Stabilizing an Application**

This unit provides an overview of the stabilizing track of a software application development project. The unit also describes the key tasks that the testers perform during the stabilizing track by using Visual Studio Team System tools.

#### **Lessons**

- Stabilization Tasks
- The Build Process
- Demonstration: Performing a Code Profiling Test
- WISP Stabilization Review Discussion
- Best Practices for Using Visual Studio Team System Tools to Stabilize an Application

#### **Lab : Stabilizing WISPLab : Project Manager**

- Defining the Security Bug Work Item Type
- Exporting the Bugs by Priority Report for GABBI

#### **Lab : Infrastructure Architect**

- Viewing Reports
- Exporting a Report

#### **Lab : Solution Architect**

- Viewing Reports
- Exporting a Report

#### **Lab : Developer**

- Viewing Reports

### **Lab : Exporting a ReportLab : Tester**

- Running Unit, Code-Analysis, and Load Tests on the WISP Web Services
- Building the WISP Web Services

After completing this unit, students will be able to:

- Project managers will be able to use Visual Studio Team System to add a work item type.
- Information architects and solution architects will be able to use the predefined reports included in Visual Studio Team System.
- Developers will be able to use the predefined reports included in Visual Studio Team System.
- Testers will be able to:
  - Run unit tests.
  - Run code-analysis tests.
  - Run load tests.
  - Log bugs.
  - Build an application.
  - Use source control.
  - Manage task assignments.

### **Session 6: Deploying an Application**

This unit provides an overview of the deployment track of a software application development project. The unit also describes the key tasks that are performed in this track by using Visual Studio Team system tools.

#### **Lessons**

- Deploying Tasks
- Components of the Final Build
- Types of Setup Projects
- Demonstration: Creating a Build Type and Building a Team Project
- WISP Deploying Review Discussion
- Best Practices for Using Visual Studio Team System Tools to Deploy an Application
- WISP Project Closing Discussion

### **Lab : Deploying WISPLab : Solution Architect**

- Branching Source Code and Creating a Final Build in WISP
- Creating a Setup Project

## **Lab : Project Manager, Infrastructure Architect, Developer, and Tester**

- Copying a Build
- Installing the Solution

After completing this unit, students will be able to:

- Solution architects will be able to:
- Branch source code files.
- Create and test the deployment build
- Project managers, infrastructure architects, developers, and testers will be able to install and run the solution.

### Prerequisites

- This workshop requires that students meet the prerequisites for each of the roles they assume:
- Project Managers
- Project managers should have two years of experience and should be familiar with process methodologies such as Microsoft Solutions Framework (MSF).
- Infrastructure Architects
- Infrastructure architects should have at least five years of IT work experience. They should have knowledge and skills in the following areas:
- Server operating systems such as Microsoft Windows Server 2000 or Microsoft Windows Server 2003.
- Networking architecture such as subnetting and routing.
- Network security including firewalls, ports, and Secure Sockets Layer (SSL).
- Infrastructure architects should be familiar with diagramming implementations, but they might have less experience with a formal diagramming methodology.
- Solution Architects

Solution architects should have at least five years of IT work experience. They should have knowledge and skills in the following areas:

- High-level database design.
- Distributed systems design.
- Business requirements analysis.
- Solution architects should be familiar with diagramming solutions, but they might have less experience with a formal diagramming methodology.
- Enterprise Developers

- Developers should have at least one to two years of experience as junior developers. They should have experience in using source code versioning control tools, bug reporting, tracking, analysis tools, and possibly a unit testing suite. Developers should:
- Have experience with Microsoft Visual Studio 2003 or Microsoft Visual Studio 2005.
- Be competent Microsoft Visual Basic .NET or Microsoft Visual C# developers.
- Software Test Engineers
- Software test engineers should have at least one to two years of experience as a software tester. They should have experience in using source code versioning control tools, bug reporting, tracking, analysis tools, and possibly a unit testing suite. They should have:
- Experience with Visual Studio 2003 or Visual Studio 2005.
- Some knowledge of Visual Basic .NET or Visual C# code.