BDC Benjamin Day Consulting

Benjamin Day Consulting, Inc.
17 Roberts St #2
Brookline, MA 02445
617-645-0188
http://benday.com

Visual Studio Team System 2008 & Team Foundation Server 2008

Visual Studio Team System 2008 (VSTS2008) helps ensure that your software development projects are successful.

Just like we all know that we should eat right and exercise regularly for our own good health, we often know what makes or breaks a software development initiative. Unfortunately, even though we know what the right thing to do is, we often don't do it. Why? Because it's hard, or it's a chore or it's inconvenient, or maybe we just forgot.

Visual Studio Team System 2008 helps your projects by making software development 'best practices' easy to do and follow.

With Team Foundation Server 2008 (TFS2008) as part of your project, you also get a single unified place for all of your project's artifacts – code, project schedule, defects, automated build information, and documents – no more wondering where something is and hunting around to find it. Since TFS2008 tracks all of these artifacts for you and tracks how they relate to one another, it can let you know whether your project is on-track or off-track and what the trend lines are.

Effective adoption of VSTS2008 begins with the developer's code. Unit tests help ensure early quality in the code base and over time help detect regressions. After the code, there's TFS2008 Source Control. TFS2008 Source Control gives you enterprise-quality version control and helps you to enforce quality and traceability by allowing you to set rules on what can be checked in.

Then there's TFS2008's Build Server and automated build functionality. Getting your official compiles off of the developer's workstation and on to a separate machine helps ensure that you *really* know if your code compiles and helps to eliminate the "works on my box" problem. Automated builds help detect quality problems early and also help you to model and manage your release cycle.

TFS2008 also gives you a way to link your code and builds to your project plan. TFS2008's Work Items help you manage your project schedule and track your defects so that you know what's done – and more importantly – what's not done in your project.

All these pieces – work items, code, source control, automated builds – can be connected via TFS2008 Linking. Linking lets you create associations between all the items in your project so that you get better traceability and to give you insight into your project's "big picture". When you check in code, you can say what task in the project plan or bug it's related to. When an automated build runs, it records which source control "check ins" and which project plan work items are in that build. If the build fails, it will automatically create a bug. This is all done via Linking.

Then finally, TFS2008 takes all the data that it's been capturing about your project and puts it into a data warehouse. The data warehouse data is displayed to the team through a suite of reports that enables you to quickly see what's happening on your project and view the overall health of your project.

Best of all, these features are customizable so that you can make TFS2008 work for you rather than you working for TFS. Together, all of these features help you and your team to follow software development best practices to deliver high quality code on-time and on-budget.

What you'll learn:

- Visual Studio Team System 2008
    - What is Visual Studio Team System and why do you care?
    - What are the pieces of Visual Studio Team System?
    - Team System Licensing
- Team Foundation Server 2008
    - Installation & Administration
        - Hardware requirements
        - How do I setup Team Foundation Server?
        - Single server or dual server?
        - How do I manage permissions in Team Foundation Server?
        - How do I connect to Team Foundation Server?
        - Team Foundation Server Proxy
    - Team Projects
        - What is a Team Project?
        - What is a Process Template?
        - Create a Team Project
        - Delete a Team Project
        - How many Team Projects should I have?
    - Version Control (Source Control)
        - TFS Version Control is not SourceSafe
        - What is a Workspace?
        - Checkin & Checkout
        - What's a changeset?
        - Shelvesets & Shelving
        - How should I set up my repository?
        - What is Branching & Merging?
        - Branching Strategies
        - Shared Checkout vs. Exclusive Checkout
        - Why is "Get Latest On Checkout" is evil?
        - Enforce code quality & traceability with Checkin Policies
        - Write a Custom Checkin Policy
        - TFS Version Control from the Command Line: tf.exe & tfpt.exe
        - Source Control Best Practices

  - Work Item Management
    - What is a Work Item?
    - Tasks, Scenarios, and Bugs
    - Areas and Iterations
    - Work Item Queries
    - Create and Edit Work Items with Excel
    - Create and Edit Work Items with Microsoft Project
    - Agile / Scrum Project Management with TFS Work Items
    - Traceability with Source Control & Builds
    - Customize your work items
    - Work Item Security
    - Get email updates from TFS using TFS Alerts
  - Automated Builds
    - Why are automated builds important?
    - Create a basic automated build
    - Continuous Integration Builds vs. Nightly Builds
    - MSBuild Scripting
    - Run unit tests from the build
    - 3rd Party MSBuild Tasks
    - Write your own MSBuild Tasks
    - Build customization
    - Using Team Build to streamline the QA process
    - Incorporate WiX into your Team Build to create MSI application installers
    - Deploy and unit test your database from an automated build
  - Reporting
    - View your project's status using the TFS Reports
    - Why is TFS Reporting important?
    - What's in the TFS Data Warehouse?
    - Facts, Dimensions, & Measures
    - Create a custom report using SQL Server Reporting Service's Report Builder
    - Access the TFS Data Warehouse from Excel
    - Customize the TFS Reports
  - Customization
    - Extend TFS using the TFS APIs
    - TFS Web Services
    - Hook in to TFS Events
    - Re-using & Extending the Team Explorer UI Controls
  - Miscellaneous
    - Team Foundation Server Power Tools
    - What's coming in Visual Studio Team System 2010?
- Features for Developers
  - Unit Testing Your Code

    - What is a unit test?
    - What is Test-Driven Development (TDD)? Why is TDD important?
    - What is the TDD process?
    - Common (bogus) excuses for NOT doing Unit Testing & TDD
    - How do I sell my team on Unit Testing & TDD?
    - Create and write unit tests
    - Architect your application for testability
    - Testing user interfaces
    - Test non-public methods
    - Strategies for unit testing legacy code
    - The RhinoMock Framework – Mocks & Stubs
    - Using Mocks & Stubs to avoid The Huge Integration Test pitfall
    - Dealing with databases and test data in your unit tests
    - Mocking web services, back-end systems, and database calls
    - Design Patterns for testability: Repository, Service Layer, N-Tier Architecture, & Model View Presenter (MVP)
    - Best Practices for fixing bugs & defects using TDD
    - Refactor for Testability
    - Refactor for Code Coverage
    - Code Profiling to identify performance problems
    - More test types: Ordered, Manual, and Generic
  - Testing Your ASP.NET Applications
    - What is a Web Test?
    - What is a Load Test?
    - Recorded or Coded Web Tests?
    - Validation Rules
    - Extraction Rules
    - Data-driven Web Tests
    - Extending the Validation and Extraction Rule Framework
  - Database Development
    - Get your database schema under source control
    - Offline Database Development
    - Unit Tests for Stored Procedures and other database objects
    - Schema Comparisons
    - Table Data Comparisons
    - Test Data Generation
    - Database Refactoring
    - Analyze your database code for ‘worst practices’
    - Deploying schema changes
    - Use the database deployment and schema compare features from a USB drive
    - Server Projects vs. Database Projects
    - Partial Projects & Composite Projects

        - The behind the scenes structure of database unit test
    - Code Quality
        - Static Code Analysis (FxCop)
        - How does Static Code Analysis work?
        - Write a custom Static Code Analysis rule
- Manage Your Project
    - Agile Methodologies
    - Scrum
    - Dealing with the hidden ‘Doneness Deficit’
    - Dealing with chaos & the unknown
    - Mitigating risk
    - Thinking Agile

Details

Technologies: Visual Studio 2008 Team System, Visual Studio 2008 Team Suite, Visual Studio 2008 Team Foundation Server, RhinoMocks, MSBuild
Programming language: C#
Duration: 5 days, 9am to 5pm
Audience: Developers & Project Managers
Cost: $2600.00 per student (includes lunch, coffee, and snacks)

Notes: The price above assumes that the student brings his or her own laptop. For public courses in the metro-Boston area, student machines can be rented for an additional $150. This course is available for on-site and private teachings. Topics covered during private sessions can be customized, compressed, and trimmed to meet your company's time constraints and training needs.

Pre-Requisites

Attendees should be familiar with basic concepts in software development methodologies, including:

- Roles in the Software Development Process. Attendees should understand basic roles such as Architect, Developer, Tester, and Project Manager.
- Configuration Management (a.k.a. Source Control). Attendees should understand the purpose and practice of configuration management as a way of preserving and reviewing project history.
- Reading code and making simple modifications by following instructions (which we'll walk you through).
- Basic features of Visual Studio 2008: toolbars, tool windows, projects, solutions, properties, and the built-in help system (position cursor, press <F1>).

VSTS/TFS Class Lab Machine Requirements

Operating System: Windows XP Professional, Windows Server 2003, Windows Vista, or Windows 7

Memory: 2GB or more

Disk: 20GB of free disk space after installation of software listed below. (Team Foundation Server lab virtual machine is approximately 15GBs.)

Other Hardware: DVD-ROM drive

Other: Students should either be an administrator on their machines or have access to an administrative account.

Required Software (in installation order):

- All available patches from Windows Update
- Microsoft Loopback Adapter (http://support.microsoft.com/kb/839013)
- Microsoft Virtual PC 2007
  - http://tinyurl.com/287fx8
- Microsoft Virtual PC 2007 Service Pack 1
  - http://tinyurl.com/6cwwoq
- Recheck for patches from Windows Update and install any available patches.

Questions? Want to talk more about this course? Want to sign up for or schedule this course? Contact us at info@benday.com