

## Developing Web Applications Using ASP .NET MVC

### Course Summary

#### Description

Since the creation of the .NET framework the recommended method for developing Web Applications has been ASP .NET Web Controls. However extending the controls metaphor to the Web has not been a complete success. Although controls enable rich user interfaces they can lead to performance problems, increased network traffic and designs that lack clear divisions between architectural layers.

The ASP .NET MVC Framework is a new approach to Web Application development, which builds on top of the existing server page infrastructure but dispenses with Web Controls. It enables a clean separation between presentation code and business logic, leading to faster development, layered architectures and components that can be unit tested.

This course provides existing developers with all the information they need to start developing ASP .NET MVC based Web Applications. The delivery is workshop based, with developers enhancing a sample application as the course progresses.

#### Topics

- The Evolution of ASP .NET MVC
- Creating a Simple MVC Web Application
- Enhancing the Web Application
- Making the Web Application Maintainable
- Combining MVC with Other Frameworks

#### Prerequisites

Students must be C# developers who are familiar with conventional ASP .NET based Web Applications. Experience with Visual Studio 2008 and AJAX is helpful but not essential.

#### Duration

Two days

## Developing Web Applications Using ASP .NET MVC

### Course Outline

- I. The Evolution of ASP .NET MVC**
  - A. Advantages and disadvantages of Web Controls
  - B. Drawbacks of the Model-View-Presenter pattern
  - C. Differentiating Web Controls from the ASP .NET infrastructure
  - D. Alternative approaches to Web Application frameworks
  - E. How ASP .NET MVC evolved as a separate product
- II. Creating a Simple MVC Web Application**
  - A. How incoming requests are routed to controllers
  - B. Declaring controllers and business methods
  - C. Capturing information from request parameters
  - D. Building and returning ActionResult objects
  - E. How an ActionResult is mapped to a view
  - F. Options for passing model information into server pages
  - G. Creating output pages using the Html helper class
- III. Making the Web Application Maintainable**
  - A. Hiding data sources using the Data Access Object Pattern
  - B. Simplifying persistence logic using the Repository Pattern
  - C. Writing Unit Tests for DAO's and Controller classes
  - D. Writing Integration and Acceptance tests
- IV. Enhancing the Web Application**
  - A. Customizing the routing information in Global.asax
  - B. Associating methods with request types via AcceptVerbs
  - C. Processing form parameters using Value Objects
  - D. Validating parameters and displaying error messages
  - E. Creating Controller Factories which perform Dependency Injection
  - F. Adding support for AJAX to server pages
- V. Combining MVC with Other Frameworks**
  - A. Using jQuery to simplify server pages
  - B. Using LINQ to enhance Data Access Objects
  - C. Writing DAO's using NHibernate and Generics
  - D. Using Unity and Spring .NET to build controllers
  - E. Creating your own View Resolvers