

ASSURING SOFTWARE QUALITY USING VISUAL STUDIO 2010

QA2010 | 3 Days



INTRODUCTION

This three-day, instructor-led course provides students with the knowledge and skills to prevent, detect, manage and avoid software defects during the application development lifecycle using Visual Studio 2010.

AUDIENCE

This course is intended for developers, testers and other QA professionals who are involved in defining, assuring and increasing the overall quality of a software application.



AT COURSE COMPLETION

After attending this course, students will be able to:

- Understand software quality assurance and quality control concepts
- Plan software quality early in the lifecycle
- Use techniques to detect and avoid defects
- Understand acceptance criteria and testing
- Understand the tester's role and responsibilities
- Test software on an Agile team
- Identify testing support in Visual Studio 2010
- Choose the appropriate Visual Studio test type
- Use Microsoft Test Manager
- Create and manage test plans, test suites, test cases and manual test steps
- Run, record and automate manual tests
- Reduce the no-repro problems with software defects
- Use the Test List, Test View and Test Results windows
- Categorize your tests using test lists, ordered tests and categories
- Re-run only those tests impacted by recent code changes
- Ensure and control the quality of .NET code through the use of unit tests, code coverage, code analysis and code metrics
- Ensure and control the quality of SQL Server code through the use of database unit tests, data generation plans and static code analysis
- Ensure and control the quality of web applications through automated web testing
- Profile an application to identify performance issues
- Ensure applications meet Quality of Service (QoS) requirements
- Use unit tests to perform acceptance testing
- Use generic tests to reuse existing testing tools
- Create, manage and report software defects
- Configure data-driven tests to cover more test cases
- Place automated tests under load
- Create Coded UI tests using several methods
- Understand Visual Studio agents, specifically test controllers and test agents
- Use check-in policies to ensure quality
- Use Team Foundation Build to perform automated builds and testing
- Enable continuous integration to support automated regression testing

PREREQUISITES

Before attending this course, students should have experience in developing and/or testing .NET, web, Windows and/or SQL Server database applications. Additionally, students should:

- Have familiarity with their organization's software development process
- Have experience reading and understanding business requirements
- Be able to understand and define acceptance criteria
- Have experience managing test cases
- Have familiarity with distributed application design (i.e. client/server, web, n-tier, etc.)
- Be able to 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 to Software Quality Assurance

This module introduces the concepts of Quality Assurance (QA) and Quality Control (QC) and how they relate to avoiding and detecting software defects. Topics include the principles around QA and QC, traditional testing approaches, test-driven development, behavior-driven development and common testing scenarios in the software development lifecycle.

Lessons

- Introduction to quality assurance and quality control
- Traditional testing
- Test-Driven Development (TDD)
- Behavior-Driven Development (BDD)
- QA and QC support in Visual Studio 2010
- Storyboard – how to plan quality
- Storyboard – how to manage test plans
- Storyboard – how to manage test cases
- Storyboard – how to trace tests to requirements
- Storyboard – how to automate testing

Demos

- Scenario 1 – Traditional testing
- Scenario 2 – Test-Driven Development
- Scenario 3 – Behavior-Driven Development

Module 2: Collaborating on Quality

This module introduces Visual Studio 2010 and its support for the software development lifecycle while illuminating those features in Team Foundation Server 2010 which can be used for collaboration by the team to assure and control software quality. Topics include team projects, work items, version control, reports, dashboards and automated builds.

Lessons

- Visual Studio 2010 editions and features
- Team Foundation Server 2010
- Team projects
- Work items
- Work item hierarchies
- Version control
- Project portal
- Reports

Lab

- Create and configure a team project
- Setup areas and iterations
- Configure version control and import legacy code
- Create user story and test case work items
- Create and execute work item queries
- Use Team Explorer, Team Web Access and Excel
- Upload and manage documents on the project portal

Module 3: Microsoft Test Manager

This module introduces the testing features and support found in the Microsoft Test Manager. Topics include managing test plans, test suites, test cases, manual test steps, executing tests, recording tests and managing test runs.

Lessons

- Testing support in Microsoft Test Manager
- Test plans
- Test suites
- Test cases
- Test settings and environments
- Managing requirements, test cases and bugs
- Running, recording and playing back tests
- Managing test runs
- Security and permissions

Lab

- Use Microsoft Test Manager to create a test plan
- Create a suite of test cases
- Create a test case and run the test
- Record a test for playback
- Configure the test environment for recording video



Module 4: Visual Studio Test Projects

This module introduces the testing features and support found in the various editions of Visual Studio 2010. Topics include an introduction to test projects, test types, test lists, running tests and managing test results.

Lessons

- Testing support in Visual Studio
- Test projects
- Test types
- Test categories
- Test lists
- Test settings
- Ordered tests
- Running tests
- Managing test results
- Test Impact

Lab

- Create a test project
- Import existing tests
- Examine and execute tests
- Evaluate test results
- Use Test Impact analysis

Module 5: Ensuring the Quality of .NET Code

This module focuses on ensuring the quality of .NET code. Topics include features found in the Professional and Premium editions of Visual Studio 2010, specifically: unit testing, code coverage, data-driven unit tests, code analysis and code metrics.

Lessons

- Testing .NET applications
- Unit testing
- Writing and running unit tests
- Testing private methods
- Data-driven unit tests
- Code coverage
- Code analysis
- Code analysis rule sets
- Code metrics
- Techniques for avoiding defects in .NET code

Lab

- Write and run various unit tests
- Configure a data-driven unit test
- Calculate code coverage
- Perform code analysis
- Compute code metrics

Module 6: Ensuring the Quality of SQL Server Code

This module focuses on ensuring the quality of SQL Server code. Topics include features found in the Premium edition of Visual Studio 2010, specifically: database projects, database unit tests, testing stored procedures, testing functions, testing triggers, writing and running database unit tests and loading test data prior to testing.

Lessons

- Testing SQL Server database code
- Database unit tests
- Writing and running database unit tests
- Testing stored procedures
- Testing functions
- Testing triggers
- Loading test data prior to testing
- Techniques for avoiding defects in SQL Server code

Lab

- Write and run various database unit tests
- Configure pre and post test scripts
- Use the Data Checksum test condition
- Use the Expected Schema test condition
- Load test data prior to testing

Module 7: Ensuring the Quality of Web Applications

This module focuses on ensuring the quality and acceptance of web applications and services. Topics include features found in the Ultimate edition of Visual Studio 2010, specifically: recording web performance tests, using extraction and validation rules and using Microsoft Test Manager to test Web applications.

Lessons

- Testing web applications
- Web performance tests explained
- Recording and running web performance tests
- Extraction and validation rules
- Data-driven web performance tests
- Parameterized web performance tests
- Coded web performance tests
- Microsoft Test Manager support

Lab

- Record a web performance test
- Add extraction and validation rules
- Drive a web test from a data source
- Generate a coded web performance test
- Use Microsoft Test Manager to test a web application

Module 8: Ensuring Quality of Service

This module focuses on ensuring the quality of service of an application in the areas of performance, load and stress. Topics include features found in the Premium and Ultimate editions of Visual Studio 2010, specifically: load tests, load testing a web test, load testing a unit test, analyzing load test results and profiling application performance.

Lessons

- Introduction to quality of service
- Load tests explained
- Creating and running load tests
- Visual Studio Agents 2010
- Test controllers and agents
- Application profiling

Lab

- Create and run load tests
- Analyze results based on different load patterns
- Profile a slow performing application

Module 9: Ensuring Software Meets Business Requirements

This module focuses on acceptance testing topics to ensure that business requirements were properly implemented. Topics include: features found in Team Foundation Server 2010 and the Microsoft Test Manager, specifically: specifying acceptance criteria, create test cases, manual tests, associating test results with requirements, using unit tests for acceptance testing, generic tests.

Lessons

- Introduction to acceptance testing
- Acceptance Test-Driven Development (ATDD)
- Associating test cases with requirements
- Using unit tests for acceptance testing
- Using unit tests for integration testing
- Using generic tests to call external programs and tests

Lab

- Define acceptance criteria and test cases
- Validate business requirements using Microsoft Test Manager
- Validate business requirements using unit tests
- Create and run a generic test

Module 10: Automating Quality Assurance and Control

This module focuses on ways Visual Studio 2010 can help assure the quality of your software automatically. Topics include: check-in policies, Team Foundation Build, build verification tests, continuous integration, regression testing, gated check-ins, private builds and setting build quality.

Lessons

- Introduction to check-in policies
- Testing check-in policy
- Code analysis check-in policy
- Policies found in Team Foundation Power Tools
- Team Foundation Build
- Creating and queuing builds
- Enabling build verification tests
- Regression testing
- Determining and settings build quality
- Continuous integration
- Private builds
- Gated check-ins

Lab

- Configure and use check-in policies
- Create and run automated builds
- Configure build verification tests
- Configure continuous integration
- Set build quality and notifications
- Setup a gated check-in build

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@dunntesting.com.