Course 339:
Implementing Test Driven Development
(2 days)

Course Description…

Test-driven development (TDD) is an evolutionary approach to development where you must first write a test that fails before you write new functional code. It was developed by Kent Beck and Ward Cunningham. It is primarily an agile approach to software development and is one of the principles of Extreme Programming. This course provides an introduction to Test Driven development and test first design.

Learning Objectives…

- Introduce the concept of development agility and the Agile Manifesto
- Review each of the major agile development methods underscoring their strengths and weaknesses
- Understand how to manage an agile environment even within a structured organizational approach
- Learn how to introduce agility into a development organization

Who should attend…

Managers, programmers, developers, executives, and anyone interested in learning the benefits of agile development.

Prerequisites…

Knowledge of current development processes, such as structured top-down development and the waterfall method.

See next page for a detailed course outline…

Course Outline…

Unit 1: General Approach to Software Development

- The Software Development Process
  - Mandatory elements
  - Transitions
- The SDLC timeline
  - Problem
  - Requirements
  - Analysis and design
  - Build
  - Test
- The Sophisticated Standard (IEEE)
- The problem with late testing
- General approaches to quality assurance

Unit 2: Basis for Test Driven Development

- Rationale for test-driven development
  - Timeboxing
  - Accommodating change
- Test First Development
- Test Driven Development
  - Specification not validation
- Benefits of TDD approaches
- Limitations of TDD
- TDD pragmatics: project size, team size, languages
- TDD and traditional testing practices

Unit 3: Application of TDD

- Rules of TDD
  - Write the test
  - Write the code
  - Refactor
- Automation of TDD
  - Framework for an integrated test (FIT)
  - Using JUNIT
- Unit and integration testing with TDD
- White box and black box testing with TDD
- Agile acceptance testing
  - Individuals and interactions
  - Customer collaboration
  - Working software
  - Response to change
- The safety net effect

Unit 4: Incorporating TDD into agile methods

- Extreme Programming
  - Working in pairs
  - Testing and coding trade-off
- Scrum
  - Working in teams
  - Incorporating a quality control role

- Agile modeling
  - TDD as a modeling approach
- Other agile approaches
- Applying TDD to structured approaches?

Unit 5: The Bottom Line

- Ideas to Use
- Where to go for more information

Please contact your ROI representative to discuss course tailoring!!!