

## **Visit – I** (Instructor is Dr. Magdy Hanna)

Basic Software Testing Track: (6 Days)

### **Course 1: Software Test Planning and Design (2 day class)**

- This course covers CSTP areas 1&2:
  - Principles of software testing
  - Test design

### **Course 2: Writing Testable Requirements (2 day class)**

- This course covers CSTP area 5:
  - Requirement definitions, refinement and verification

### **Course 3: Software Inspections & Reviews: A process Oriented Approach (2 day class)**

- This course covers CSTP area 7:
  - Static testing ( Inspections, reviews and walkthroughs)

## **Visit – II** (Instructor is Ms. Marnie Hutcheson)

Basic Software Testing Track: (4 Days)

### **Course 1: Managing the Testing Process, Test Execution and Defect Tracking (2 day class)**

- This course covers CSTP areas 3&4:
  - Managing the testing process
  - Test executions and defect tracking

### **Course 2: Software Test automation: Principles, Architectures, and Techniques. (2 day class)**

- This course covers CSTP area 6:
  - Basic test automation

# VISIT I

## Outlines and Workshop Overviews:

### Course #1:

## Software Test Planning and Design

*(Two-Day Session)*

This course covers CSTP Areas #1 and #2:

#### Principles of Software Testing

- Levels of Testing
- Testing client/server
- Testing Mainframe applications
- Testing Internet and web applications
- Testing object-oriented applications
- Testing embedded systems
- The testing life cycle

#### Test Design

- Code-based test case design techniques
- Requirement-based test case design techniques
- Test design specification

### Course Overview

#### Why This Course

This course is essential to all software professionals, programmers, test analysts, engineers, development managers and test managers. This course offers a systematic approach to effective software testing.

*Software Test Planning and Design* is for those interested in:

- Performing effective unit testing
- Gaining more control over the different test activities
- The quality of the software product,
- Performing effective planning and design
- Monitoring all test activities.

This course offers a systematic approach to effective software testing.

Whether you have been doing testing for a number of years or you have just started, you need this course. You will learn a number of testing practices that you might not have seen before. The course will also cover the different types of testing performed at each phase of the software lifecycle and issues involved in these types of tests. The course will also cover the test design phase of the testing lifecycle. It starts with in-depth coverage of test case design techniques both for black and glass box testing. This includes Equivalence Class Partitioning, Boundary Value Analysis, Cause-Effect Diagram, Decision Tables and Decision Trees. The last part of the course will cover test design activities and writing Test Design Specifications including documenting test conditions, test cases, test scripts and procedures, and expected results.

## **Course Outline:**

### **Principles of Software Testing**

- Testing defined and terminology
- Requirements
- Test condition
- Test cases
- Test scripts/procedures
- Testware relationships
- Blackbox/Glass Testing

### **Disciplined Software Testing Practices**

- Requirements are crucial for effective testing
- Test Both functional and quality requirements
- Adopt Model-Based requirements
- Formally design your tests
- Perform both positive and negative testing
- Trace requirements to test components
- Trace test cases to database components
- Perform thorough regression testing
- Define your test process
- Select tools to support your process
- Perform both static and dynamic testing
- Continue formal education

### **Testing Levels**

- Unit testing
- Integration testing
- System testing
- Major areas of systems testing
- Usability and GUI testing
- GUI Navigation Diagrams
- User acceptance
- Regression testing
- The structured testing Pyramid
- The object-oriented testing pyramid

### **System Test Planning**

- Why plan?
- Developing a test strategy
- Test documentation

- Components of a test plan
- A test plan template
- Creating a systems test plan
  - Identification of the test plan
  - Systems test environment
  - Systems test objectives and scope
  - Systems test approach
  - Systems test staffing and responsibilities
  - Hardware/software/network requirements
  - Testing tools
  - Systems test deliverables
  - Systems test tasks

### **Test Case Design Techniques**

- Strategies for generating test cases
- Black Box testing
  - Equivalence Class Partitioning
  - Boundary Value Analysis
  - Cause-Effect Diagram
  - Decision Tables
- Glass Box testing
  - Statement/decision/condition coverage
  - Path coverage
  - Program complexity and basis path coverage

### **Systems Test Design**

- Test design basics
- Test design activities
- Deliverables of the test design phase
- Conditions and test cases
- Procedures and expected results
- Major areas of system testing
- Business requirement testing
- Preparing the Test Design Specification
  - Defining test conditions
  - Testing for missing business requirements
  - Defining test cases
  - Identifying data needed to execute tests
  - Defining baseline data
  - Determining expected results
  - Defining test cycles
- A Systems Test Specification template

# VISIT I

## Course #2:

# Writing Testable Requirements and Requirement-Based Testing

*(Two-Day Session)*

This course covers CSTP Area #5:

### Requirement Definitions, Refinement and Verification

- Writing testable requirements
- Exploring requirements
- Refining requirements
- Defining requirements
- Requirement verification
- Requirement tractability

## Course Overview

### Why this course:

This two-day course is essential for test and quality professionals, requirement engineers, business and system analysts and everyone who is involved in writing, validating and testing requirements. It covers details on how to document different types of requirements in a form that is testable. The course adopts a model-driven requirement process. It is based on the instructor's philosophy of using models to assure completeness, correctness, testability, and precession of requirements. The course will also discuss the elements of requirement management process.

You will learn:

- Requirements without models is simply waste of time and resources
- Learn how models can tremendously improve the testability of requirements
- Learn how to identify test scenarios for model-based requirements
- Learn the elements of a requirement management process and learn how to build your own process

**Course Outline:**

1. Requirement Validation
  - Types of requirements
    - Requirement traceability
    - Functional Vs quality requirements
    - Documenting quality requirements
    - Assessing the quality of software requirements
    - Forms of requirements
2. The Common Sense Model for Software Development and Testing
3. The Requirement Management Process
4. Models for Exploring and Refining Requirements
  - Data models
  - Process models
  - Use Cases
  - State Models
  - User Interface Models

# VISIT I

## Course #3:

# Software Inspections and Reviews: A Process Oriented Approach

*(Two-Day Session)*

This course covers CSTP Area #7

### **Static Testing (Inspections, Reviews, and walkthroughs)**

- Types of static testing
- The process of static testing
- Defect data analysis
- Improving the process

### **Course Overview**

#### **Why this course:**

Software inspections have proved to be very effective in capturing more defects early enough to avoid the cost of rework. Also, many types of defects can only be found by inspection. Yet, many organizations are struggling to make the inspection process effective and to achieve the real benefits of inspections. This course will examine the different elements that make an effective inspection process and help you establish a process, which is customized for your organization. This course is a must for all software persons involved in the software development and maintenance process including managers of all levels. You will learn:

- What the real benefits of inspections are
- How to achieve the most value of your inspection effort
- The different factors affecting the inspection process
- The different forms of inspection and when to use each of them
- How to establish different process variations to fit different environment
- How to effectively monitor your inspection process to improve its effectiveness
- Most importantly, learn from instructor's real experience with inspections

## Course Outline:

- ❑ Introduction
  - Why do we need to inspect
  - Benefits of inspection
  - Why do inspection programs fail
  - Inspection after or before unit testing
  - Different forms of inspection
  - Formal reviews
  - Internal review cycle
  - Walkthroughs
  - Formal inspections
  - Verification and validation
  
- ❑ Establishing the Process
  - Why do we need a process
  - Characteristics of effective process models
  - Steps to establish the process
  
- ❑ Gathering Relevant Information
  - Development process
  - Types of systems developed
  
- ❑ In-class Team Exercise
- ❑ Defining a Standard Inspection Process 6. Defining Process Variations
- ❑ Training
- ❑ Monitoring
- ❑ Elements of An Inspection Process
- ❑ Planning
  - Planning the inspection
  - Selecting the moderator
  - Selecting the inspectors
  
- ❑ The Overview meeting
- ❑ Preparation
- ❑ The Inspection Meeting
  - How to detect lack of preparation
  - What happens during the inspection meeting
  - Rules of the inspection meeting
  
- ❑ Record Keeping and Recording
- ❑ Rework and Follow-up
- ❑ Causal Analysis

# VISIT II

## Outlines and Workshop Overviews:

### Course #1:

## Managing the Testing Process and Test Execution and Defect Tracking

*(Two-Day Session)*

This course covers CSTP Areas #3 and #4:

#### Managing the Testing Process

- Planning
- Scheduling
- Reporting
- Resources
- Risk Management
- Measuring and improving the test process

#### Test Executions and Defect Tracking

- Test scripting
- Reporting
- Defect tracking

### Course Overview

#### Why This Course?

This two day seminar is based on the book “Software Testing Fundamentals,” authored by the presenter (Published by Wiley, May 2003), and in part on the author’s new book in progress. The Seminar is a field guide aimed squarely at testers and test managers. It contains practical solutions, to real life testing challenges. This flexible methodology can add value to your test effort whether you are involved in a traditional plan driven test effort or a bleeding edge Agile / eXtreme effort.

The seminar presents a “Best Practice” risk based methodology for planning, sizing, managing and executing the test effort that is both defensible and reproducible. Case studies illustrate how to tailor the methods to get the best results in traditional plan driven, RAD, Rapid Application Development, and Agile / eXtreme efforts. These test methods are applicable whether you are testing Internet/Intranet applications, traditional client server applications, mobile application environments, or closed systems and embedded

firmware environments. The seminar is filled with real world examples of how testers successfully managed their test efforts and demonstrated the value of testing.

## Course Outline:

- **Day 1: Risk Based Test Management**

- Background and Concepts

- Introduction
- Fundamental methods and metrics for software and testing applied to the 3 major development types
  - Traditional Plan Driven Development
  - Rapid Application Development (RAD)
  - Agile / eXtreme Development
- Test estimation and sizing
  - The Test Inventory

- Tools and analysis techniques for test estimation and planning

- Identify and Enumerate Tests, fill out the Inventory, prioritize the tests, identify the most important tests, estimate the effort, track the test effort and evaluate it
- Analysis techniques for enumerating tests and test environments
- Risk analysis
  - Establishing Test Priority
  - Relating risk to test coverage
  - Requirements
- Finishing the Test Estimation Effort
  - Answering the question: How big is it?
  - Setting schedules, and goals
  - Estimating the resources and time required to test the most important tests
  - Negotiating for resources to conduct the test effort
- Adding value beyond the test effort

## Day 2: Identifying and Executing the Most Important Tests

- **Identifying the Most Important Tests**

- Introduction: Analytical Methods for Performing Risk Based Testing
- Analytical Methods for picking the Most Important Test
  - Path Analysis
  - Applied Path Analysis
  - Data Analysis
  - Test Selection
    - Applying Risk analysis and Ranking to Selected tests

- **Executing the Most Important Tests**

- Managing Test Execution and Reporting Test Progress
  - S-Curves – How they work
  - S-Curves in Practice
- Reporting and Tracking Bugs
  - Case Study: A Successful Agile Effort
  - Analyzing Bug Statistics
  - The time-to-fix versus risk “Z” index
- Measuring the performance of the test effort and demonstrating the value of testing

# VISIT II

## Course #2:

# Principles Architectures and Strategies of Test Automation

*(Two-Day Session)*

This course covers CSTP Area #6:

### Test Automation

- Tool evaluation and selection
- Architectures
- Automation standards and guidelines
- Planning the test automation process
- Automation team roles

## Course Overview

### Why this course:

This course provides an introductory approach to test automation. The objective of this class is to provide the student with a general knowledge of how to incorporate test automation into current test practices. The student is provided knowledge on how to get started, develop successful test automation strategies for short and long term ROI, and manage the test automation expectations of testers, developers, managers and business clients. Items covered will include:

- Awareness of test automation stakeholder expectations
- Test automation doesn't just include automating test cases alone
- Importance of test automation constructs that are customized to your processes and development culture
- Learn how to modularize your test automation into reusable and maintainable components
- Learn through real life examples to demonstrate some of the concepts
- Learn how customized your test automation software development into time and cost saving solutions
- This class will build your confidence in dealing with the day-to-day decisions of test automation development.

This course is valuable to anyone considering test automation, in the process of implementing test automation, or who is looking to improve the test automation they have already implemented. Although prior experience and knowledge of automation scripting would be nice, it is not required. The audience is any stakeholder that would benefit from the investment of automation. This would include but is not limited

to Business Analysts, Test Managers and Practitioners, Developers, Project Managers, CIOs, CTOs and Business Application Owners.

## **Course Outline:**

### **Introduction**

- What is Software Testing?
- How does Test Automation Play a Role in testing?
- What skills sets are required for test automation?
- Who should be involved in tool selection?
- Different levels of expectations
- Planning & Communication
- Measurements of success

### **Automation Framework**

- Test Requirements file creation
- Creating & maintaining test automation documentation
- Logging test results for easy analysis and reporting
- Keeping test automation useable in a high-change environment

### **Workflow Testing**

- Simplify code maintenance
- Create reusable navigational, entry actions, and test data functions.
- Utilize Excel workbooks to document & drive tests

### **Class project(s)**

- Group Assignment #1 – Break into groups to generated test automation development requirements.
- Group Assignment #2 – Break into groups to identify stakeholders and how to best get test automation started within your organization

### **Review Session**