EFFECTIVE METHODS OF SOFTWARE TESTING
WORKSHOP

Who Should Attend:

This three-day workshop is a general software testing workshop covering a wide range of testing techniques across the system life cycle. The techniques, methods and tools presented may be applied to: single stand alone personal computers, local and wide area networks, client/server multi-platform and stand alone mainframe systems. This includes the application in a multi-programming language environment. Therefore, any professional involved with or related to systems planning, analysis, requirements definition, design, construction, testing, or maintenance of software will find the content of value. Business analysts, quality assurance and audit personnel as well as key user/clients will also find this workshop most beneficial.

Course Description:

This workshop provides a comprehensive coverage of the testing processes available to support the development, delivery and enhancement of quality software. The focus is on integrated testing processes and procedures emphasizing the functional part of testing which should be an integral part of the software development and support process with reasonable costs. The course covers the principles, the processes and the documentation of software testing, verification and validation.

Almost half of the session is devoted to early testing procedures. Here, test processes supporting analysis, requirements definition, and logical design are covered. This material provides a cohesive method for avoiding the early errors which are extremely costly to correct if they are not discovered until after coding or code generation. The methodology supports large and or small projects. It covers how to manage and perform testing processes better following proven methods utilizing the best tools all the way through development. It also covers enhancements and regression testing. The philosophy here is, “If you have code and then consider testing, you are late!”

The remainder of the session concentrates on more traditional testing topics. Test plans, scenarios, scripts and test case design is covered in detail, along with function and process testing, module, unit, integration, system, and user acceptance testing, build/test strategies, test data bases, and thread testing. The measurement of testing effectiveness, and the strategies for software unit, integration, system, and regression test specification, design, and implementation are presented. Different methods are compared and the process stopping criteria are discussed along with the economics of various testing options.

Special attention is directed at early verification techniques such as the functional part of testing, requirements specification and design inspection, definition of the operational profile of the product, and at automated testing and availability of support tools for testing. Finally, issues of test completeness, test measurement, and the tuning of testing strategies are covered.
COURSE OBJECTIVES:

Upon completion of the EFFECTIVE METHODS OF SOFTWARE TESTING workshop participants should be able to:

1. establish clear scope, objectives and well-defined requirements for software testing.

2. apply appropriate software testing tools, techniques and methods for even more effective systems during both the test planning and test execution phases of a software development project.

3. understand and execute the necessary software testing steps per type of system(s), platform(s) and program(s) to be developed or enhanced.

4. know and perform the process for system test planning, execution and reporting which validates that the system meets requirements.

5. follow an effective, step-by-step process for identifying and designing test plans, scripts, scenarios and cases, and building and executing them.

6. manage and/or perform the necessary methods of software testing even more effectively applying proven methods and tools for effective quality assurance, quality control, audit and documentation of software-based systems.

MAJOR TOPICS TO BE COVERED:

1. Background & Introduction

2. Software Testing the State of the Ark

3. The Different Methods, Levels & Types of Software Testing

4. Setting Measurable Test Goals & Objectives

5. The Traditional Systems Development Life Cycle (SDLC) & Key Software Testing Points:
   - The Phases of Development and Key Testing Actions
   - Identifying and Defining Functional System Requirements To Be Tested
   - Where & When & What Test Plans Need to be Constructed
   - Where & When & What Test Data Needs to be Developed
   - Where & When Software Testing Needs to be Conducted

6. Different Types of System Life Cycles & Software Testing Strategies per Type:
   - Module and Unit Testing
   - Integration Testing
   - System Testing
   - Volume Testing
   - User/Client Acceptance Testing
   - Sign-off & Production

- Requirements, Item Conditions and Population Analysis
- Characterizing Test Conditions Using the Various Transaction Types
- Developing Test Conditions & Expected Results
- Truth Tables: Decision Tables & Testing
- Code Coverage
- Test Scripting, Etc.

8. Maintenance/Enhancement and Regression Testing

9. Testing Packaged Software

10. Prototyping & Contracting for Software—Testing Considerations

11. Testing Additional Types of Systems, Platforms and Languages

12. Recording the Results of Testing


14. Developing Test Reports

15. Mandatory Checklists, Forms, Logs & Reports for Effective Software Test Planning & Execution


17. Automated Aids for Software Testing

18. Procedures for Identifying Critical Software Attributes

19. User/Client Driven Computing Testing Considerations

20. People & Resource Considerations

21. Management Considerations for Software Testing & Reliability

22. Quality Assurance, Quality Control & Audit Considerations

23. Library & Documentation Considerations

24. New Standards for Software Testing

25. New Techniques & Future Considerations

26. Getting Started: Building and/or Updating Your Own Software Testing Methodology and Developing An Action Plan

27. Review & Evaluation