

Certified Software Tester™ (CST)

Common Body of Knowledge

- *Control Procedures*
- *Problem Resolution*
- *Reports*
- *Requirements*
- *Test Builds*
- *Test Cases*
- *Test Execution*
- *Test Plans*
- *Test Planning*
- *Testing Concepts*
- *Testing Deliverables*
- *Validation & Verification*

Class Room Materials Provided

- *CST Student Work Book*
- *CST Practice Test Questions*
- *Access to Online Prep Exam Site*

The Certified Software Tester Program™ (CST) is appropriate for individuals who want to enter the testing field or experienced Quality Assurance/Tester professionals seeking certification. The program focuses on the concepts and application of the essential testing principles that are required to perform industry standard quality assurance practices

The purpose of the program is to provide the pertinent knowledge and skills for effective Quality Assurance testing in the client/server and web environments and, to describe the essentials of Quality, Quality Assurance (QA) and Quality Control (QC). There is a special emphasis on verification & validation and their relationship to testing. Topics include such areas as test case execution on GUI and Web applications; documenting and controlling testing activities with requirements trace ability matrices, test plans, test procedures and test cases; configuration and management of online defect tracking applications, categorizing, tracking and documenting software errors and, building management reports.

Core Topics

- Essentials of the Quality Assurance Practice
- Principles of Testing
- Test Documentation Techniques

Target Audience:

If you are exploring new career opportunities or want to stay current in your field, SofTest's certification program can provide focused career training and skill development for working professionals, and recent college graduates. Our certification programs are designed to supplement degree education. Qualified software managers, consultants, and software engineers teach the classes.

- Career Change
- Quality Assurance Analysts, Testers, or leads that are updating their testing skills and need to enhance managerial skills
- Anyone interested in improving the quality and maintainability of their software

Prerequisites:

Basic computer skills, capability to access and navigate the Internet, familiarity with executing software applications, previous exposure to client/server, web, and email concepts and familiarity with the Windows environment.

Duration:

Day Class: 5 sessions

CST Curriculum:

Course 1: Essentials of the Quality Assurance Practice

Module 1: Quality Assurance (QA) & Quality Control (QC) Practice

This module provides the fundamental concepts and terminologies related to Quality Assurance and the relationship between Quality Assurance, Quality Control and Testing. In addition, this module covers the benefits of the overall Quality Assurance process for both the development and testing teams. It expounds on why the essential knowledge and discipline is required for all phases of the development life cycle.

You Will Be Able To:

- Discuss the primary objectives of Quality Assurance
- Explain the essential activities of Quality Assurance
- Distinguish the difference between Quality Assurance and Quality Control
- Define Quality and Quality Testing
- Discuss the verification and validation aspects of testing

Main Topics Covered:

- The Three Principles of Quality
- Quality in Information Services
- Quality Assurance (QA) and Process
- Quality Control (QC) and Process
- Quality Testing
- The Roles of QA/QC
- QA/QC Test Planning

Module 2: Systems Development & Testing Life Cycles

This module introduces the Systems Development Life Cycle (SDLC), Traditional Project Life Cycle and the Product Management Life Cycle. It covers the phases, tasks, inputs, deliverables or outputs for each Life Cycle in detail. It also identifies the relationship between the Testing Process, Testing Life Cycle and the SDLC, and how, in some instances, it can overlap multiple phases of the SDLC.

The module also covers the initiation and review process of systems projects.

You Will Be Able To:

- Explain the purpose of a mission statement
- Explain the SDLC structured approach to designing and building application systems

- Describe how system projects begin and how projects are reviewed
- Discuss the various methods used to develop information systems
- Explain the overall Testing Life Cycle and how it covers multiple phases of the SDLC
- Describe quality assurance and test activities performed in typical system development projects during the SDLC phases
- Discuss the Traditional Project Life Cycle phases, tasks, and deliverables
- Describe Product Management Life cycle phases, inputs and outputs

Main Topics Covered:

- Systems Development Life Cycles (SDLC)
- Traditional Project Life Cycle
- Product Management Life Cycle
- Systems Development Life Cycle and Testing Life Cycle
- Quality Assurance & Test Process Activities
- Test Planning Activities
- Test Preparation Activities
- Unit Test Activities
- System Test Activities
- Integrated System Test Activities
- Acceptance Test Activities
- On-Going Maintenance

Module 3: Tracking Software Changes

This module describes issues and terminology related to tracking software changes. In addition, it explores why managing software changes is a critical part of the systems development and quality assurance functions. Change control, scope creep, version control and configuration management are also discussed in detail.

You Will Be Able To:

- Define “Change Control” and how it is used to monitor and manage software changes
- Define “Scope Creep” and identify ways to prevent it
- Explain how “Version Control” manages multiple versions of software applications
- Describe “System Configuration Management” procedures

Main Topics Covered:

- Software Changes
- Tracking Software Changes
- Change Control
- Scope Creep

CERTIFIED TESTER PROFESSIONAL PROGRAM

- Version Control
- Configuration Management

Module 4: Reviews and Inspections

This module examines quality assurance objectives for Reviews and Inspections; and the noticeable differences between the two. Also covering the various types of reviews and inspection forms and checklists. Reviews and Inspections are among the most valuable tools available for quality assurance and play a significant role in the quality management process when used consistently and correctly.

You Will Be Able To:

- Explain the evolution of the inspection process
- Explain what inspections are designed to accomplish
- Explain who uses inspections and why
- Understand the value that occurs when inspections are used
- Explain the similarities between reviews and inspections
- Name the various types of reviews
- Explain what reviews are designed to accomplish
- Discuss “review” critical success factors
- List and describe test products that can be reviewed
- List and discuss the roles and responsibilities of review participants
- Discuss the basic steps of the reviews and inspections processes and their objectives

Main Topics Covered:

- Reviews & Inspections – Differences, Roles & Basic Process Steps
- Differences Between Reviews & Inspections
- The Inspection Process
- Quality Assurance Inspection Objectives
- Inspecting the Software Procedures Manual
- Software Requirements
- Inspection Meeting Agenda
- The Review Process
- Types of Reviews
- Components of the Review Plan
- Critical Success Factors for Reviews
- Review Checkpoints

Course 2: Principles of Testing

Module 1: Fundamentals of Testing

This module provides the history and fundamentals of testing with specific emphasis on Software Testing. It also

explores the various strategies, categories and types of testing that can occur within the testing process. Generally, testing is the method of executing a program with the specific objective of finding errors. A successful test therefore is a test that uncovers undetected errors.

In addition, this module examines the two major objectives of the testing process - designing and executing, defined by measurement goals, data collection, software reliability and quality.

You Will Be Able To:

- Explain the history and purpose of testing
- Define the basic testing principles
- Explain the key testing issues
- Define the three major testing organizations and explain their test methodologies
- Explain the different testing strategies and how they are used
- Explain the different testing categories, how they are used and by whom
- Explain the different testing types and how they are used
- Explain the difference between User Interfaces (UI) and User Interface Controls

Main Topics Covered:

- Six Principles of Testing
- Testing is Fundamental
- Testing Practices and Strategies
- Black Box Testing
- Unit Level Testing Model
- Integration Level Testing Model
- System Level Testing Model
- Acceptance Level Testing Model
- What is a User Interface (UI)?
- User Interface Controls
- Types of Interface Controls
- Windows Compliance Testing (Checklist)
- Considerations for Testing User Interaction
- Checklist for a Good Interface
- Considerations for Testing Input Basics, Navigation, and Viewing Operations
- Considerations for Testing Windows Interface Components
- Considerations for Testing Menus, Shortcuts and Controls

Module 2: Defects and Problem Reporting

This module expounds on the primary goals of the tester, which is to identify, classify and report errors as early as

possible in the development life cycle. An error (issue) occurs when a program does not operate in the manner in which its user expects during the software development life cycle.

You Will Be Able To:

- Define the primary objective of the problem reporting process
- Explain the fundamental principles of reporting defects
- Identify when and understand why problem reports should be submitted
- Explain what items are included in a problem report
- Explain why software has defects
- Define the major categories of software errors
- Explain the concept of severity and priority as they relate to defects
- Explain the basic features of the Elementool defect (bug) tracking tool

Main Topics Covered:

- Tracking and Reporting Defects
- Reporting a Software Problem
- Submitting Problem Reports
- Problem Report Content
- Major Categories of Software Errors
- Severity of Defects
- Priority by Severity

Module 3: Automated Testing

This module elaborates on how automated testing tools help organizations optimize and accelerate the delivery of applications. It also addresses the paradox in purchasing, setting-up and executing an automated test environment.

In addition, this module reviews automated testing methods that allow for the quick capture and reuse of quality activities to share and repeat throughout the testing life cycle.

You Will Be Able To:

- Explain the key functions of an automated test tool
- Understand the advantages and disadvantages of using an automated tool
- Understand the issues surrounding test automation
- Understand what to automate and what not to automate

Main Topics Covered:

- Concepts of Automated Testing
- Why Automate Testing
- Functions of Automated Test Tools
- Disadvantages of Using Automated Testing Tools
- Automated Tool Issues
- Guidelines for What to Automate

Course 3: Test Documentation Techniques

Module 1: Project Documentation

This module introduces Project Documentation and its criticality within the software development process and the multiple purposes it serves.

You Will Be Able To:

- Explain the importance of project documentation
- Define the goals of documentation standards
- Explain the various project documentation categories

Main Topics Covered:

- Project Management Documentation
- Software Quality Assurance Documentation
- Software Configuration Management Documentation
- Software Verification and Validation Documentation
- Requirements Documentation
- Design and Implementation Documentation
- Test Documentation
- User Documentation
- Installation and Checkout Documentation
- Operation and Maintenance Documentation

Module 2: Requirements Documentation

This module focuses on the two main reasons for having Requirements Documentation. The first is that the information contained within the documented requirements drives the Systems Development Life Cycle activities and these activities are essential for the software development and Quality Assurance processes. The second is that the outcome of the requirements definition is a documented agreement between the customer and the development team that explicitly describes the product to be developed.

You Will Be Able To:

- Define what a “requirement” is and its importance to the Quality Assurance process

CERTIFIED TESTER PROFESSIONAL PROGRAM

- Explain why requirements are important
- Explain what a Requirement Document is and the importance
- Explain what a Software Requirements Specification (SRS) Document is and its importance
- Explain how and where requirements fit into the Software Development Life Cycle (SDLC)
- Define characteristics that make requirements testable
- Explain how a Requirements Traceability Matrix is used and its importance

Main Topics Covered:

- Requirement Definition and Document
- Software Requirements Specification (SRS) Document and Checklist
- Checklist of Words and Grammatical Constructs Prone to Ambiguity
- Functional / Nonfunctional Requirements
- Tracking Requirements and Traceability Matrix
- The Impact of Poorly Documented Requirements
- Requirements Impact on the Software Development Life Cycle
- Characteristics of Testable Requirements
- Guidelines for Writing Requirements

Module 3: Requirements-Based Test Methodology

This module focuses on requirements and their relationship to the testing and-test process. It also explains how requirements incorporate test conditions and functional checklists to verify correct system functions.

You Will Be Able To:

- Explain what Software Inspections are, how they are used, and why they are used
- Explain what Prototypes are and how they are used
- Define Black Box testing and explain how users and non-users use it
- Explain what a Use Case is and how it is used

Main Topics Covered:

- Software Prototypes
- Test Requirements
- How Do Test Requirements Relate To The Test Plan
- Entrance & Exit Criteria To Generate Test Requirements
- Exit Criteria For Test Requirements
- Black Box Methodology
- Non-User Black Box Categories
- Advantages and Disadvantage of Black Box Testing
- Black Box Testing Strategies and Techniques

- Use Case Concepts and Techniques
- General Guidelines for Completing a Use Case

Module 4: Industry Standard Test Documentation

This module discusses test documentation as a tool and how it is used to administer and maintain the testing process along with how it aids in the planning, monitoring, and managing of the testing phases. This module also reviews the three primary layers of test documentation: Test Plan Definition, Test Plans, and Test Cases.

You Will Be Able To:

- Understand the importance of Test Documentation
- Understand how to develop a Test Plan
- Define the different types of Test Plans
- Explain the differences between a Test Plan and a Comprehensive Test Plan
- Explain the types of things that are included in a Test Plan Definition document
- Define what a Test Case is and how it is used
- Explain the differences between Test Cases and Test Scripts

Main Topics Covered:

- Test Plan Definitions and Standards
- How to Develop a Test Plan
- IEEE Standard for Test Documents to the Execution Process
- Relationship of Test Documents to the Execution Process
- Test Case and Script Definition and Standards
- How Test Cases Are Generated
- IEEE Standards for Test Cases
- Characteristics of a Good Test Case
- Basic Test Case Guidelines
- Test Case Specification
- Test Case Log Forms
- Test Execution Log Forms
- Checklists Methodology