DIGILEAF INC.
Leading Excellence Among Fellows

Certified Software Quality Professional

IAQP and DigiLEAF advocate that the value of a certification program is in direct proportion with the ability to meet individuals' various needs and interests for both breadth and depth of content so that it helps them improve the way they perform their roles and responsibilities in their respective organizations.

This education-based certification program is designed by industry and academic practitioners to professionalize the software quality practices in the country. It has been seen that competence levels of those in the field of software quality have to grow to meet the growing demands of the industry to produce quality products and services.

The Certification Program in Software Quality is designed to cater to practitioners specializing in the areas of Software Quality Assurance and Software Testing (Quality Control). After taking the program, an individual shall demonstrate high competence levels both in concepts and practical applications that are required for a professional software quality practitioner.

After taking the training program, individually, each participant shall be able to:

1. Integrate tools, techniques, and strategies in software quality and use these in IT-related projects.
2. Apply quality principles and methodologies in IT-enabled processes.

The knowledge areas used in the curriculum are aligned with the bodies of knowledge used in other international certifications like those offered by ISTQB, IIST, AST, ASQ, QAI.

Course Curricula

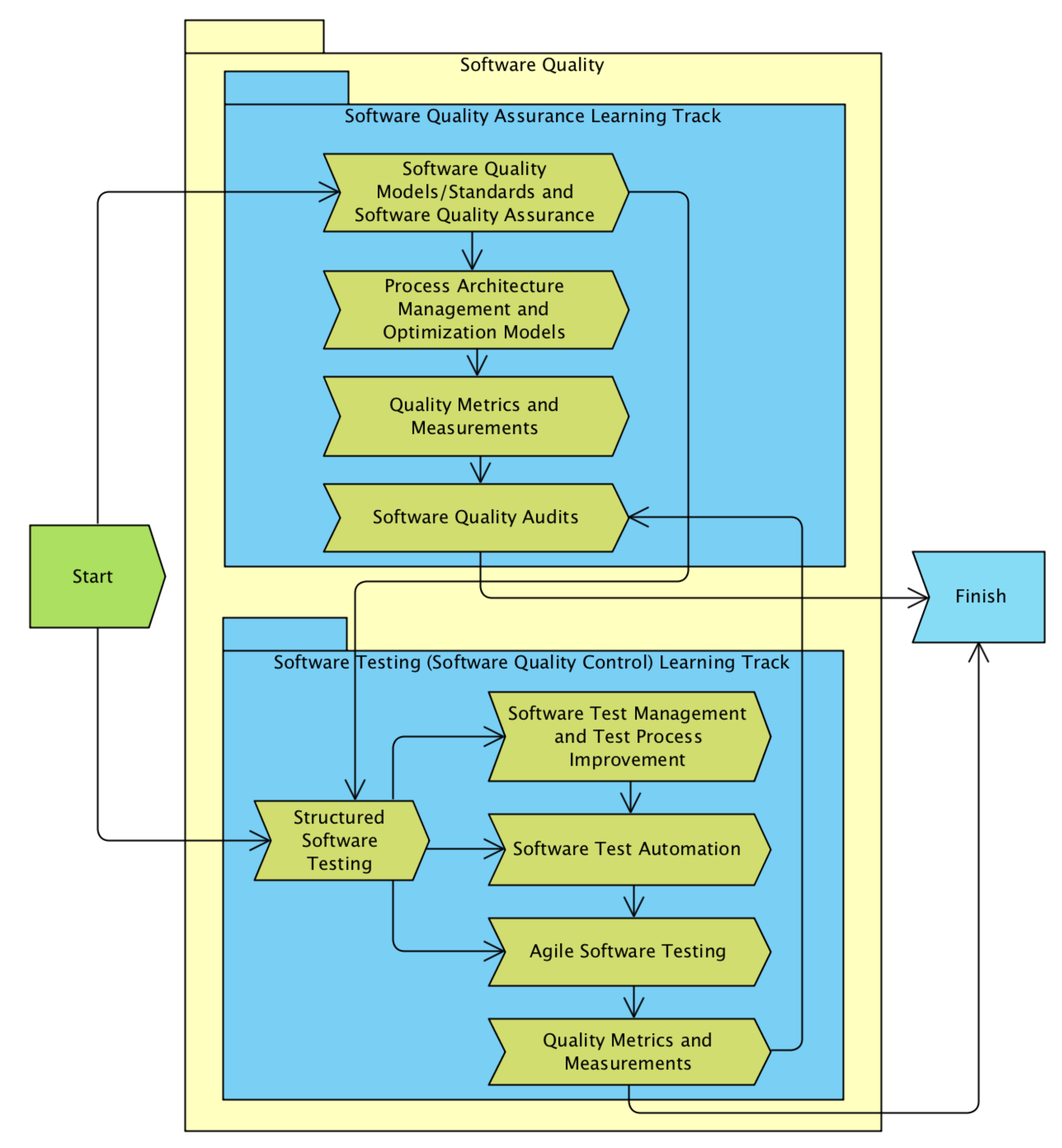
Training Program Duration: 16 Days
Review & Mock Exam: 2 days | Certification Exam: 0.5 days

1. Software Quality Models/Standards and Software Quality Assurance
2. Process Architecture Management and Optimization Models
3. Structured Software Testing
4. Software Test Management and Test Process Improvement
5. Quality Metrics and Measurements
6. Software Test Automation
7. Agile Software Testing
8. Software Quality Audits

DIGILEAF INC.

Leading Excellence Among Fellows

Course Map



Software Quality

Software Quality Assurance Learning Track

Software Quality Models/Standards and Software Quality Assurance

Process Architecture Management and Optimization Models

Quality Metrics and Measurements

Software Quality Audits

Start

Finish

Software Testing (Software Quality Control) Learning Track

Software Test Management and Test Process Improvement

Structured Software Testing

Software Test Automation

Agile Software Testing

Quality Metrics and Measurements

Course Description

Software Quality Models/Standards and Software Quality Assurance

SQ-001, Duration: 2 Days

This course is an introduction to software quality models & standards. It presents best practice guidance on how to apply an international quality standard to the acquisition, supply, development, operation and maintenance of computer software. The course discussion is independent of the technology, life cycle models, development processes, sequence of activities and organizational structure used by an organization.

Process Architecture Management and Optimization Models

SQ-002, Duration: 2 Days

This course sets the baseline of process-orientation and opens up the world of process management and process optimization. This course will cover other major approaches aside from BPR to provide wide knowledge on what particular approach/methodology best fits an organization. This course will also present and examine the challenges, benefits and risks of each approach. A strategic framework for developing a business transformation roadmap and planning process change will help the attendees be prepared to manage change efforts in their businesses.

Structured Software Testing

BA-004, Duration: 2 Days

This course presents how to test software based on requirements in a risk-driven approach regardless of project type. It discusses all necessary documents to be used as inputs to software testing and the documents to be developed within the testing processes. Techniques and approaches to test case development are discussed in detail. Test results captured in test status reporting are presented in a structured manner. This course includes practical skills necessary to effectively comprehend requirements documented in a systems requirements specification to be used as one of the inputs in writing test plans and test cases.

Quality Metrics and Measurements

SQ-006, Duration: 2 Days

This course discusses in detail how to setup and manage test metrics. Writing of test measurement plan and tracking process is covered in detail. Identifying critical test metrics would enable the test team to provide objective decisions in the evaluation of the system under test.

Certified Software Quality Professional

Software Test Management and Test Processes Improvement

BA-005, Duration: 2 Days

This course presents the management of the testing process in the context of software engineering. It covers the application of quality risks and how to apply a risk-driven test development. The participants would be able to understand the value of testing in making a “go/no-go” decisions in software deployment.

This course covers a modern approach in test process improvement. This includes the roles and responsibilities associated for test process improvement project. Gap analysis and next steps to undergo to proceed to the next level of process maturity is presented.

Software Test Automation

SQ-007, Duration: 2 Days

This course will discuss a step-by-step process to determine if an organization is prepared for test automation. It encompasses the principles used for automated test effort preparation, tool selection, and formation of an automated test team with workable applications and examples in scripting methods, test implementation, assessment and measurement for automated test efforts. It stresses on acquiring the most out of automated testing tools, error handling and logging methods, and developing an efficient automated testing architecture.

Agile Software Testing

SQ-008, Duration: 2 Days

This course covers the concepts, practices and implementation of agile software testing. After knowing the standard artifacts needed in testing software projects, participants will learn how to strategize, plan, design and execute tests in short development iterations and with incomplete specifications.

Software Quality Audits

SQ-009, Duration: 2 Days

This course is intended to audit software related processes and procedures, software products, and the people who perform the software processes and procedures. General audit practices will be discussed. The audit will now depend on what standard will be used as a requirement of the audit. Therefore, the course is not concentrated on specific standards like ISO or CMMI, etc. The auditing techniques that will be learned from this course are standard audit practices. Specific auditing techniques will be covered to be used as a value-added information when software/IT standards are used as requirements.

DIGILEAF INC.

Leading Excellence Among Fellows

Certified Software Quality Professional

Training Investment

Course Code	Course Title	Pre-requisite	Duration (Day)	Regular Public Training Fee per Course/Pax	Certification Package Fee
SQ-001	Software Quality Models/Standards and Software Quality Assurance	None	2	P16,661.12	P84,519.60
SQ-002	Process Architecture Management and Optimization Models	None	2	P16,661.12	
SQ-004	Structured Software Testing	None	2	P16,661.12	
SQ-005	Software Test Management and Test Processes Improvement	SQ-004	2	P16,661.12	
SQ-006	Quality Metris and Measurements	None	2	P16,661.12	
SQ-007	Software Test Automation	SQ-004	2	P16,661.12	
SQ-008	Agile Software Testing	SQ-004	2	P16,661.12	
SQ-009	Software Quality Audits	None	2	P16,661.12	
	Review and Mock Exam		2	P16,661.12	Free
	Certification Exam		0.5	P17,200	Free
			18.5 days	**P167,150.08**	**P84,519.60***

**Total Discount from Regular Public Training Fees: P82,630.48 (49%)*

Certified Software Quality Professional

International Alliance of Quality Professionals (IAQP)

Certification Levels

Level 1: Competence Assessed

Title: Certified Software Quality Professional [CSqP]

Graduate of any four-year degree course or Minimum of three years experience in the software quality field for non-degree holders. Proficient in written and oral English

Level 2: Principles Applied in a Project

Title: Certified Software Quality Professional [CSQP]

The candidate should demonstrate the required competences and aptitude as described within the Certified Software Quality Professional (CSqP) Level 1 or any other international certification body offering certifications in software quality (e.g. ISTQB, IIST, AST, ASQ, QAI). The said certification shall be submitted to the steering committee for approval and endorsement for level II.

Upon endorsement and assessment, the candidate shall submit a letter of intent to apply for CSQP with the intended project case. The project case must demonstrate practical application of the body of knowledge. The candidate shall have 6 months to accomplish the project case and request for assessment. A reviewing board shall then be formed to assess the completion of the project through document reviews and face-to-face interviews.

Upon successful completion and justification, the reviewing board shall certify that the candidate has attained level II certification.

Level 3: Managed Software Quality (QA or Test) Professionals in an organization

Title: Certified Software Quality Professional, Fellow [CSQP, Fellow]

The candidate should demonstrate the required competences and aptitude as described within the Certified Software Quality Professional (CSqP) Level 1 or any other international certification body offering certifications in software quality (e.g. ISTQB, IIST, AST, ASQ, QAI). The said certification shall be submitted to the steering committee for approval and endorsement for assessment to level III.

The candidate should demonstrate the practical application of the body of knowledge through a project case as certified by a reviewing committee as endorsed by DigiLEAF Inc.

The candidate should demonstrate the required competences and aptitude as indicated in the ASQ Certified Manager of Quality/Organizational Excellence.

Target audience: Anyone working within the roles of quality assurance, software testing and quality management.

DIGILEAF INC.
Leading Excellence Among Fellows

Software Quality Models/Standards and Software Quality Assurance

This seminar/workshop discusses the paradigm of software quality and how it is used as a driver for business excellence. Technology has to be aligned with business to fully appreciate the value it brings to an organization. Appropriate use of quality tools and techniques are covered to properly equip any professional who will focus on applying system/software quality. Discussions include software quality assurance and software testing practices and how these disciplines could be tailor-fitted in an organization regardless of software development methodology.

Training Objectives

At the end of the course, the participants will be able to:

1. Describe international quality standards to software and IT environments.
2. Identify various software quality models to be used in a technology-driven organization.
3. Integrate software quality models for maximum benefits.
4. Define a clear strategy for setting up or managing software quality management system.
5. Differentiate the software development group, software quality assurance group, software testing group.
6. Describe SQA standards.
7. Discuss the business benefits of having an independent SQA and Test Group in an organization.

Topics

I. Seeing the Whole Value Chain for Quality Deployment

II. Levels of Quality Management Maturity
   a) Level 1 – 5
   b) Characteristics of each level
   c) Standards/Tools/Techniques at each level

III. Paradigm of Continuous Improvement
   a) Lean
   b) Six Sigma
   c) Lean Six Sigma
   d) Juran RCCA
   e) JDI

IV. Thrust of Software Quality management

V. Software Quality Assurance Processes

VI. Tailor-fitting Software Quality Assurance Processes aligned with Software Development Processes

VII. Software Quality Control Processes (a.k.a. Software Testing)

VIII. Defining Appropriate Types and Levels of Tests to Various Applications

IX. Continuous Improvement as applied to the Software Industry

Duration **2** days

DIGILEAF INC.
Leading Excellence Among Fellows

Process Architecture Management and Optimization Models

This course covers comprehensive principles and applications of process management. This course discusses techniques and methods to fully equip anyone who will be involved in problem solving or quality improvement initiatives. This course sets the baseline of process-orientation and opens up the world of process management and process optimization. A strategic framework for developing business transformation roadmap and planning process change are presented.

Training Objectives

At the end of the course, the participants will be able to:

1. Determine appropriate process management approaches, preventive and corrective actions.
2. Use process analysis, design, review and documentation tools/techniques.
3. Prepare policies, process maps and procedure documents.

Duration **2** days

Topics

I. Process Management Concepts
   a) Various Meanings of Quality
   b) Quality Maturity Model
   c) Definitions of Process Terms
   d) Principles of Process Management
II. Using SIPOCO Diagram
   a) How to model a SIPOCO diagram
   b) Value of a SIPOCO diagram
III. Process Analysis, Design & Documentation
   a) Importance of Process Mapping
   b) Process Mapping Using flowcharts
   c) Process Mapping Using UML – Activity Diagram
   d) Concepts of Policies/Procedures
   e) Elements of a Policy Statement
IV. Problem Taxonomies
   a) Chronic vs. Sporadic Problem
   b) Types of Quality Problems
V. Principles involved in various Problem Solving Methodologies
VI. 6S: A Plan for Neat and Clean Workplaces
VII. Lean Kaizen in the 21st Century
   a) Kaizen
   b) Poka-yoke (Mistake Proofing)
   c) Mistake Proofing the Process
   d) Poka-yoke's in Software

DIGILEAF INC.
Leading Excellence Among Fellows

Structured Software Testing

This course presents how to test software based on requirements in a risk-driven approach regardless of project type. It discusses all necessary documents to be used as inputs to software testing and the documents to be developed within the testing processes. Techniques and approaches to test case development are discussed in detail. This course includes practical skills necessary to effectively comprehend requirements documented in a systems requirements specification to be used as one of the inputs in writing test plans and test cases.

Training Objectives

At the end of the course, the participants will be able to:

1. Define basic testing concepts based on the new ISO/IEC/IEEE 29119 test standard.
2. Develop a focused, risk-based test plan to prioritize what to test and manage the test project.
3. Develop test cases based on functional and non-functional requirements.
4. Use the new ISO/IEC/IEEE 29119 standard in writing test plans, test cases and test procedures.
5. Apply verification and validation strategies.

Duration **2** days

Training Outputs

1. Quality Risk Analysis (using Failure Mode and Effect Analysis)
2. Test Plans
3. Test Cases
4. Test Procedures

Topics

I. Software Testing Concepts
   a) What is Testing?
   b) Why Do We Need To Test?
   c) Goals of Testing
   d) Static vs. Dynamic Testing
   e) Verification vs. Validation
   f) Exhaustive Testing
   g) Testing as a Heuristic

II. The Test Process Based on ISO/IEC/IEEE 29119
   a) The Test Process Model
   b) Organizational Test Process
   c) Test Management Process
   d) Dynamic Test Process
   e) Test Development Lifecycle
   f) Relationship Between Project and Test Sub-Project

III. Risk-Based Testing: Quality Risk Analysis
   a) What is a Quality Risk
   b) What To Do With Quality Risks as it Relates to Testing?

IV. Test Planning
   a) Introduction to Test Planning
   b) What is the ISO/IEC/IEEE Standard 29119?
   c) Roles in Software Testing
   d) Allocating Testing Resources: The Test System
   e) Levels of Test and Types of Tests
   f) Aligning Testing in the Overall Quality Directions
   g) Re-testing and Regression Testing

V. Test Strategies and Tactics
   a) Analytical Test Strategies
   b) Model-Based Test Strategies
   c) Methodical Test Strategies
   d) Process-Oriented Test Strategies
   e) Dynamic Test Strategies
   f) Philosophical Test Strategies
   g) Regression Test Strategies
   h) Common Testing Challenges/Problems

VI. Test Design Concepts
   a) What is a Test Case?
   b) Elements of a Test Case
   c) Test Case Specifications
   d) Test Procedure Specifications
   e) Writing Effective Test Cases
   f) Fundamental Testing strategies
      i. Black-box Testing
      ii. White-box Testing
      iii. Gray-box Testing

VII. Test Case Design Approaches/Techniques
   a) Validation Methods
   b) Equivalence Partitioning
   c) Boundary Value Analysis
   d) Scenario Based Testing

VIII. Preparing Test Data
   a) Test Data Set Categories
   b) Dummy Data vs. Business Data

Software Test Management and Test Processes Improvement

This course presents the management of the testing process in the context of software engineering. It covers the application of quality risks and how to apply a risk-driven test development. The participants would be able to understand the value of testing in making a "go/no-go" decisions in software deployment.

This course covers a modern approach in test process improvement. This includes the roles and responsibilities associated for test process improvement project. Gap analysis and next steps to undergo to proceed to the next level of process maturity is presented.

Training Objectives

At the end of the course, the participants will be able to:

1. Apply quality risk analysis, determine test project scope, and prepare a budget and schedule.
2. Institute and use bug and test tracking systems that support effective test reporting.
3. Prepare for and manage the challenges of test execution.
4. Build and retain an effective test team.
5. Work and communicate effectively with managers and peers, especially when communicating test findings.
6. Improve testing processes.
7. Make visible step-by-step improvements for a fast payback.
8. Describe various Test Process Improvement models.
9. Implement improvements in conjunction with or independent of the Capability Maturity Model Integration (CMMI).

Topics

I. Pragmatic Testing Unravelled
   a) Effective and Efficient Testing
   b) Right Level of Efficiency
II. Aligning Testing in the Project
III. Test Strategies, Tactics, and Design test strategies
   a) Analytical
   b) Model-based
   c) Methodical
   d) Process-oriented
   e) Dynamic
   f) Philosophical
   g) Regression
   h) Three Other Regression Strategies
   i) Test Tactics
   j) Categories of Testing Techniques
   k) Strategic and Tactical Considerations
IV. Managing Test Execution and Reporting Test Progress
V. Applying Process Improvement to the Testing Process
VI. Using the Test Maturity Model (TMM)
VII. How to Conduct TMM Assessments
VIII. Structured Test Process Improvement

Duration **2** days

DIGILEAF INC.
Leading Excellence Among Fellows

Quality Metrics and Measurements

Ability to measure the right things is imperative in these modern times. Even successful businesses find that they also need to assess other aspects of their business not just financial performance. This course is designed to equip the participants in designing metrics that are specific, measurable, achievable, realistic & time-bound. In this course, participants will be able to prevent misguided metrics; infer what data is needed and how to collect it; use a proven process for designing metrics and evaluate metrics' effectiveness. This course also covers how to quantify process performance and results in software quality. This course discusses methods and tools to gather, analyze and interpret metrics and measurement software engineering, software testing processes and other related process areas.

Training Objectives

At the end of the course, the participants will be able to:

1. Identify the reasons for measurement.
2. Determine elements to measure.
3. Define various models in measurement.
4. Align performance measures in objectives
5. Setup a measurement plan in a project.
6. Use common graphical tools for metrics presentation and reporting.

Duration **2** days

Topics

I. Introduction to Metrics and Measurement
   a) Rationale Needed in Measurements
   b) Scales of Measurement
   c) Characteristics of a Good Measure
   d) Measure vs. Metric

II. Performance Measurement Design Process
   a) Understanding and Mapping Business Structures and Processes
   b) Stakeholder Analysis
   c) Using Critical-to-Quality (CTQ) in Stakeholder Analysis
   d) The CTQ Tree
   e) Developing Business Performance Opportunities
   f) The Kano Model
   g) Understanding the Current Performance Measurement System
   h) Developing Business Performance Indicators
   i) Using Goal-Question-Metric (GQM) Model in Deriving Indicators
   j) Using Input-Process-Output-Outcome (IPOO) in Deriving Indicators
   k) Deciding How to Collect the Required Data
   l) Designing, Reporting and Performance Data Presentation Formats
   m) Testing and Adjusting the Performance Measurement System
   n) Implementing the Performance Measurement System

III. Metrics in Software Testing and Software Engineering Processes

DIGILEAF INC.
Leading Excellence Among Fellows

Software Test Automation

This course will discuss a step-by-step process to determine if an organization is prepared for test automation. It encompasses the principles used for automated test effort preparation, tool selection, and formation of an automated test team with workable applications and examples in scripting methods, test implementation, assessment and measurement for automated test efforts. It stresses on acquiring the most out of automated testing tools, error handling and logging methods, and developing an efficient automated testing architecture.

Training Objectives

At the end of the course, the participants will be able to:

1. Identify necessary factors for successful test automation.
2. Identify what testing segment can be automated in each phase of software development.
3. Evaluate automated testing tools.
4. Develop and utilize a valid and consistent strategy for planning, developing, executing, reporting and maintaining automated tests.
5. Estimate the cost and expectations on investment returns in terms of efficiency gains and bugs shipped.
6. Set realistic test automation goals.
7. Recognize the required talents in an automation test team.
8. Establish test automation activities.

Duration 2 days

Topics

I. Introduction to Test Automation
   a) What Automation Can Achieve and What It Costs
   b) Building the Right One and Avoiding the Bad

II. Automation Standards and Principles
   a) Automated Testing Lifecycle Methodology
   b) Build and Utilize a Test Framework

III. Tool Evaluation and Selection
   a) Selecting the Right Tool for the Task
   b) Build or Buy?
   c) Management Commitment
   d) Product Support and Education

IV. Architectures – Test Assets Arrangement for Efficiency
   a) Test Asset Management
   b) Test Scripts/Cases as Organization Assets
   c) Scale, Reuse, Compound Versions, Platform Dependence/Independence

V. Test Automation Process Preparation
   a) Initial Factor to Automate
   b) Capture versus Create
   c) Scope Cost and Timelines Approximation
   d) Automation Team Roles

VI. Automated Test Effort Implementation
   a) Test Construction
   b) Test Execution
   c) Test Results Capture & Analysis
   d) Test Results Verification

Agile Software Testing

This course covers the concepts, practices and implementation of agile software testing. After knowing the standard artifacts needed in testing software projects, participants will learn how to strategize, plan, design and execute tests in short development iterations and with incomplete specifications.

Training Objectives

At the end of the course, the participants will be able to:

1. Apply the concepts of Agile development.
2. Test effectively in agile project environments.
3. Describe the agile life cycle.
4. Describe the agile development practices.
5. Perform exploratory testing.
6. Strategize testing in an agile environment.

Duration **2** days

Topics

I. Understanding the Agile Mindset
   a) The Agile Manifesto
   b) Agile Values
   c) Agile Principles
II. Introduction to Scrum
   a) What is Scrum?
   b) The Scrum Framework
   c) Scrum Principles
   d) Scrum Aspects
   e) Scrum Processes
III. Commonly Used Agile Artifacts
IV. Agile Testing Strategies
   a) Defining Agile Testing
   b) The Role of Agile Testers
   c) Testing in an Agile Environment
   d) Agile Techniques & Concepts
   e) Testing in the Agile SDLC
   f) Testing within an Iteration
   g) Agile Testing Quadrants
V. Testing during Feature Development
   a) Agile Development Lifecycle
   b) Agile Development Methodologies
   c) Developer testing versus user testing
   d) Feature unit testing strategies
VI. Exploratory Testing
   a) Exploratory testing, defined
   b) Context-Driven School
   c) Session-Based Exploratory Testing
   d) Ad-hoc vs. Exploratory Testing
   e) Elements of SBET
   f) What is Chartering?
   g) Elements of a Charter
   h) Recommendations for Good Chartering

Software Quality Audits

This course covers how to audit processes, work products, and the people who perform within the process. General audit practices will be discussed. The audit will now depend on what standard will be used as a requirement of the audit. Therefore, the course is not concentrated on specific standards like ISO or CMMI, etc. The auditing techniques that will be learned from this course are standard audit practices. Specific auditing techniques will be covered to be used as a value-added information when software/IT standards are used as requirements for audit.

Training Objectives

At the end of the course, the participants will be able to:

1. Identify the activities involved in quality auditing.
2. Describe audit roles.
3. Write a software audit plan.
4. Prepare audit tools.
5. Document the results of an audit in an audit report.
6. Write a corrective action plan.
7. Prepare quantifiable recommendations for improvements.

Duration 2 days

Topics

I. Auditing Principles
   a) Audit Definitions
   b) Reasons for Auditing
   c) Types of Audits
II. 1st, 2nd & 3rd Party Audits
   a) What Software Audit Should Do?
   b) Audit Triggers
   c) Audit Phases
      i. Initiation
      ii. Preparation
      iii. Execution
      iv. Reporting
      v. Follow-Up
III. Audit Planning
   a) Elements of an Audit Plan
   b) Audit Basis
   c) Auditors Toolkit
   d) Developing a Protocol or Checklist
IV. Audit Execution
   a) The Audit Triangle
   b) Vertical vs. Horizontal Auditing
   c) Audit Strategy
      i. Tracing Method
      ii. Discovery Method
      iii. Element Method
      iv. Department Method
   d) Audit Analysis
      i. Corroboration and Objectivity of Evidence
      ii. Data Patterns and Trends
      iii. Audit Findings
V. Corrective Actions
VI. Audit Reporting
VII. Follow-up Audit
VIII. Auditing Continual Improvement