LAPT

LONDON ACADEMY OF PROFESSIONAL TRAINING

CERTIFICATE

# ISO 15026 — Systems and Software Assurance

*ISO Certification Programme*


**LAPT — London Academy of Professional Training**

**Address** 85 Great Portland Street, W1W 7LT, London, United Kingdom

**Email** info@lapt.org

**Telephone** +44 7513 283044

**Web** lapt.org

**Reference** IIT-SFT-15026 • ISO IT & Related Technologies

UKPRN 10067351 • Registered in England and Wales, company 4984798

| 6 | 30 | 180 | 430 | 60% | Lifetime |
|---|---|---|---|---|---|
| SUBJECTS | CHAPTERS | CLASSES | TOTAL MARKS | PASS MARK | VALIDITY |

## AT A GLANCE

| | |
|---|---|
| REFERENCE | **IIT-SFT-15026** |
| AWARD | **Certificate** |
| ASSESSMENT | **430 marks — 258 to pass (60%)** |
| PREREQUISITES | **None** |
| VALIDITY | **Lifetime** |
| AWARDING BODY | **LAPT — London Academy of Professional Training** |

## CURRICULUM

### 1 ISO 15026 Standards Application

5 chapters · 30 classes **105 marks**

- **1 Understanding ISO 15026: Foundations and Key Principles** — 6 classes
  - 1.1 Define ISO 15026 and Its Importance in Assurance
  - 1.2 Explore the Core Principles of ISO 15026 Standards
  - 1.3 Identify Key Components of Systems and Software Assurance
  - 1.4 Discuss the Role of Leadership in Implementing ISO 15026
  - 1.5 Analyze Case Studies on ISO 15026 Application
  - 1.6 Develop a Plan for Integrating ISO 15026 into Projects
- **2 Framework and Structure of ISO 15026 Standards** — 6 classes
  - 2.1 Understand the Purpose of ISO 15026 Standards
  - 2.2 Explore the Key Components of the ISO 15026 Framework
  - 2.3 Analyze the Structure of ISO 15026 Documentation
  - 2.4 Identify Roles and Responsibilities within ISO 15026
  - 2.5 Compare ISO 15026 with Other Relevant Standards
  - 2.6 Apply ISO 15026 Principles to Real-World Scenarios
- **3 Implementing ISO 15026: Best Practices and Methodologies** — 6 classes
  - 3.1 Assessing Current Systems Against ISO 15026 Standards
  - 3.2 Identifying Best Practices for Software Assurance
  - 3.3 Developing a Step-by-Step Implementation Plan
  - 3.4 Integrating ISO 15026 with Existing Processes
  - 3.5 Training Teams on ISO 15026 Methodologies
  - 3.6 Measuring Success: Evaluating ISO 15026 Implementation
- **4 Assurance Case Development within ISO 15026** — 6 classes
  - 4.1 Define the Components of an Assurance Case
  - 4.2 Identify Stakeholder Needs for Assurance Cases
  - 4.3 Develop Argument Structures for Assurance Cases
  - 4.4 Gather and Analyze Evidence for Assurance Cases
  - 4.5 Organize Assurance Case Documentation Effectively
  - 4.6 Review and Validate Assurance Cases for Compliance
- **5 Evaluating Compliance and Continuous Improvement in ISO 15026** — 6 classes
  - 5.1 Identify Key Requirements of ISO 15026 for Compliance
  - 5.2 Assess Current Practices Against ISO 15026 Standards
  - 5.3 Develop Metrics for Measuring Compliance in ISO 15026
  - 5.4 Analyze Non-Conformities and Their Impact on Assurance
  - 5.5 Implement Continuous Improvement Strategies Based on ISO 15026 Findings
  - 5.6 Review and Adapt Compliance Processes for Future Efficiency

## 2 Leadership and Team Management

5 chapters · 30 classes **45 marks**

- **1 Understanding Leadership Theories in Software Development** — 6 classes
  - 1.1 Explore Key Leadership Theories in Software Development
  - 1.2 Analyze the Impact of Transformational Leadership on Teams
  - 1.3 Compare and Contrast Transactional vs. Servant Leadership Approaches
  - 1.4 Assess the Role of Situational Leadership in Project Management
  - 1.5 Identify Leadership Styles that Foster Team Collaboration
  - 1.6 Develop a Personal Leadership Strategy for Software Teams
- **2 Building High-Performing Teams for Software Projects** — 6 classes
  - 2.1 Identify Key Traits of High-Performing Teams
  - 2.2 Assess Team Dynamics and Roles in Software Projects
  - 2.3 Implement Effective Communication Strategies
  - 2.4 Foster a Culture of Collaboration and Trust
  - 2.5 Create Measurable Team Goals and Objectives
  - 2.6 Evaluate Team Performance and Continuous Improvement
- **3 Effective Communication and Collaboration in IT Teams** — 6 classes
  - 3.1 Explore the Importance of Communication in IT Teams
  - 3.2 Identify Barriers to Effective Team Communication
  - 3.3 Develop Active Listening Skills for Better Collaboration
  - 3.4 Utilize Tools for Enhanced Team Communication
  - 3.5 Implement Feedback Mechanisms to Foster Team Engagement
  - 3.6 Create a Collaborative Action Plan for Team Success
- **4 Managing Conflicts and Fostering Innovation in Teams** — 6 classes
  - 4.1 Identify Common Sources of Conflict in Teams
  - 4.2 Analyze Conflict Resolution Strategies for Effective Leadership
  - 4.3 Develop Communication Skills for Navigating Team Tensions
  - 4.4 Foster a Culture of Innovation through Collaborative Problem Solving
  - 4.5 Implement Techniques for Encouraging Diverse Perspectives in Team Discussions
  - 4.6 Measure the Impact of Conflict Management on Team Performance and Innovation
- **5 Evaluating Team Performance and Growth in Agile Environments** — 6 classes
  - 5.1 Assessing Team Dynamics in Agile Frameworks
  - 5.2 Identifying Key Performance Indicators for Agile Teams
  - 5.3 Conducting Effective Team Retrospectives
  - 5.4 Implementing Continuous Feedback Mechanisms
  - 5.5 Facilitating Growth through Coaching and Mentoring
  - 5.6 Measuring Success: Tools and Techniques for Agile Performance Evaluation

## 3 Principles of Systems and Software Assurance

5 chapters · 30 classes **85 marks**

- **1 Foundations of Systems and Software Assurance** — 6 classes
  - 1.1 Define Key Concepts in Systems and Software Assurance
  - 1.2 Analyze the Importance of Assurance in Software Development
  - 1.3 Identify Common Risks Associated with Software Systems
  - 1.4 Explore the Roles and Responsibilities in Assurance Processes
  - 1.5 Evaluate Assurance Frameworks and Standards
  - 1.6 Apply Best Practices for Implementing Assurance Strategies
- **2 Quality Management Principles in Assurance** — 6 classes
  - 2.1 Understand Quality Management Principles in Software Assurance
  - 2.2 Explore Key Components of Quality Management Systems
  - 2.3 Analyze the Role of Leadership in Quality Assurance
  - 2.4 Apply Quality Management Tools and Techniques
  - 2.5 Evaluate Quality Assurance Practices against ISO 15026 Standards
  - 2.6 Develop a Quality Management Action Plan for Software Projects
- **3 Risk Management Strategies for Software Assurance** — 6 classes
  - 3.1 Identify Key Concepts in Risk Management for Software Assurance
  - 3.2 Analyze Common Risk Factors Affecting Software Assurance
  - 3.3 Evaluate Risk Assessment Techniques for Effective Software Management
  - 3.4 Develop Risk Mitigation Strategies for Software Projects
  - 3.5 Implement Monitoring and Review Processes for Risk Management
  - 3.6 Apply Real-World Case Studies to Enhance Risk Management Practices
- **4 Assessment and Evaluation Techniques for Assurance** — 6 classes
  - 4.1 Identify Key Assessment Techniques for Systems Assurance
  - 4.2 Analyze the Role of Evaluation in Software Assurance
  - 4.3 Compare Quantitative and Qualitative Assessment Methods
  - 4.4 Implement Risk Assessment Tools for Software Evaluation
  - 4.5 Develop a Framework for Continuous Assurance Assessment
  - 4.6 Create Action Plans Based on Assessment Outcomes
- **5 Leadership Roles in Systems and Software Assurance** — 6 classes
  - 5.1 Define Leadership Roles in Systems Assurance
  - 5.2 Identify Key Responsibilities of Software Assurance Leaders
  - 5.3 Explore Effective Communication Strategies for Assurance Leaders
  - 5.4 Analyze Case Studies of Leadership in Software Assurance
  - 5.5 Develop a Leadership Action Plan for Assurance Practices
  - 5.6 Evaluate Leadership Styles and Their Impact on Assurance Outcomes

## 4 Project Evaluation and Improvement

5 chapters · 30 classes **45 marks**

- **1 Understanding Project Evaluation Frameworks** — 6 classes
  - 1.1 Define Project Evaluation Frameworks
  - 1.2 Explore Key Components of Evaluation Frameworks
  - 1.3 Analyze Common Evaluation Models in Projects
  - 1.4 Identify Benefits and Challenges of Project Evaluation
  - 1.5 Apply Evaluation Frameworks to Real-World Scenarios
  - 1.6 Develop a Tailored Evaluation Plan for a Project
- **2 Key Performance Indicators and Metrics** — 6 classes
  - 2.1 Define Key Performance Indicators for Project Success
  - 2.2 Identify Quantitative and Qualitative Metrics in Projects
  - 2.3 Analyze the Importance of Baseline Measurements
  - 2.4 Develop a Framework for Tracking Project Performance
  - 2.5 Evaluate Project Outcomes Using KPI Data
  - 2.6 Formulate Action Plans for Continuous Improvement Based on Metrics
- **3 Qualitative and Quantitative Assessment Techniques** — 6 classes
  - 3.1 Differentiate Between Qualitative and Quantitative Assessments
  - 3.2 Identify Key Metrics for Quantitative Evaluation
  - 3.3 Explore Techniques for Gathering Qualitative Data
  - 3.4 Analyze Sample Data Using Quantitative Methods
  - 3.5 Compare and Contrast Qualitative Findings with Quantitative Analysis
  - 3.6 Develop an Integrated Assessment Framework for Project Improvement
- **4 Continuous Improvement in Project Management** — 6 classes
  - 4.1 Analyze Current Project Performance Metrics
  - 4.2 Identify Areas for Improvement Using Root Cause Analysis
  - 4.3 Develop Action Plans for Continuous Improvement
  - 4.4 Implement Feedback Loops for Ongoing Evaluation
  - 4.5 Measure the Impact of Improvement Initiatives
  - 4.6 Foster a Culture of Continuous Improvement in Project Teams
- **5 Reporting and Communicating Project Evaluation Results** — 6 classes
  - 5.1 Analyze Project Evaluation Metrics Effectively
  - 5.2 Summarize Key Findings from Project Evaluations
  - 5.3 Create Engaging Visual Reports for Stakeholders
  - 5.4 Tailor Communications to Diverse Project Audiences
  - 5.5 Incorporate Feedback into Project Improvement Plans
  - 5.6 Present Project Evaluation Results Confidently

## 5 Quality Assurance and Control Techniques

5 chapters · 30 classes **65 marks**

- **1 Understanding Quality Assurance Principles in Software Development** — 6 classes
  - 1.1 Define Quality Assurance Principles in Software Development
  - 1.2 Explore Key Quality Assurance Standards and Frameworks
  - 1.3 Analyze the Role of Quality Assurance in the Software Development Lifecycle
  - 1.4 Identify Common Quality Assurance Techniques and Their Applications
  - 1.5 Develop a Quality Assurance Plan for Software Projects
  - 1.6 Evaluate the Impact of Quality Control on Software Quality Outcomes
- **2 Implementing Quality Control Techniques for Software Projects** — 6 classes
  - 2.1 Identify Quality Control Techniques for Software Projects
  - 2.2 Analyze the Importance of Quality Metrics
  - 2.3 Implement Testing Strategies for Effective Quality Control
  - 2.4 Monitor Compliance with Quality Standards
  - 2.5 Conduct Root Cause Analysis to Improve Quality
  - 2.6 Evaluate the Effectiveness of Quality Control Techniques
- **3 Risk Management and Mitigation Strategies in Quality Assurance** — 6 classes
  - 3.1 Identify Key Risks in Quality Assurance Processes
  - 3.2 Analyze Risk Impact on Project Outcomes
  - 3.3 Develop Effective Risk Mitigation Strategies
  - 3.4 Implement Risk Management Frameworks in QA
  - 3.5 Monitor and Evaluate Risk Mitigation Effectiveness
  - 3.6 Create a Risk Management Action Plan for QA Projects
- **4 Tools and Metrics for Measuring Software Quality** — 6 classes
  - 4.1 Identify Key Metrics for Software Quality Assessment
  - 4.2 Analyze Common Quality Assurance Tools and Their Uses
  - 4.3 Evaluate the Role of Automated Testing in Quality Control
  - 4.4 Implement Performance Metrics for Continuous Improvement
  - 4.5 Compare Qualitative and Quantitative Metrics in Software Projects
  - 4.6 Develop a Software Quality Metrics Dashboard for Project Tracking
- **5 Continuous Improvement and Leading QA Practices** — 6 classes
  - 5.1 Identify Key Concepts of Continuous Improvement
  - 5.2 Analyze the Role of Leadership in Quality Assurance
  - 5.3 Implement Effective QA Practices in Teams
  - 5.4 Evaluate Common Quality Assurance Tools and Techniques
  - 5.5 Develop a Continuous Improvement Action Plan
  - 5.6 Measure and Assess Outcomes of QA Practices

6

## Risk Management in Software Assurance

5 chapters · 30 classes **85 marks**

- **1 Fundamentals of Risk in Software Assurance** — 6 classes
  - 1.1 Define Key Concepts in Risk Management
  - 1.2 Identify Types of Risks in Software Assurance
  - 1.3 Analyze the Impact of Risks on Software Projects
  - 1.4 Evaluate Risk Assessment Techniques and Tools
  - 1.5 Develop a Risk Mitigation Strategy for Software Assurance
  - 1.6 Review and Communicate Risk Management Plans
- **2 Identifying and Assessing Risks in Software Projects** — 6 classes
  - 2.1 Define Key Terms in Risk Management
  - 2.2 Identify Common Risks in Software Projects
  - 2.3 Explore Risk Assessment Techniques
  - 2.4 Analyze Case Studies of Past Software Failures
  - 2.5 Develop a Risk Assessment Matrix
  - 2.6 Create a Risk Mitigation Plan for a Software Project
- **3 Risk Analysis Techniques for Software Assurance** — 6 classes
  - 3.1 Identify Key Risks in Software Assurance
  - 3.2 Analyze Impact and Likelihood of Software Risks
  - 3.3 Utilize Qualitative Risk Assessment Techniques
  - 3.4 Implement Quantitative Risk Assessment Methods
  - 3.5 Prioritize Risks for Effective Management
  - 3.6 Develop Risk Mitigation Strategies for Software Projects
- **4 Developing Risk Mitigation Strategies** — 6 classes
  - 4.1 Analyze Risk Factors in Software Assurance
  - 4.2 Identify Key Risks and Their Impact
  - 4.3 Develop Effective Risk Mitigation Objectives
  - 4.4 Create Actionable Mitigation Plans
  - 4.5 Prioritize Risks for Mitigation Strategies
  - 4.6 Evaluate the Effectiveness of Mitigation Strategies
- **5 Monitoring and Reviewing Risk Management Processes** — 6 classes
  - 5.1 Identify Key Risk Indicators in Software Assurance
  - 5.2 Analyze Current Risk Management Processes
  - 5.3 Develop Effective Monitoring Strategies for Risks
  - 5.4 Review and Evaluate Risk Management Effectiveness
  - 5.5 Implement Continuous Improvement in Risk Practices
  - 5.6 Communicate Risk Findings to Stakeholders Effectively