

LAPT

LONDON ACADEMY OF PROFESSIONAL TRAINING

CERTIFICATE

ISO 25010 — Systems and Software Quality Models

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-COD-25010 • ISO IT & Related Technologies

UKPRN 10067351 • Registered in England and Wales, company 4984798

AT A GLANCE

REFERENCE	IIT-COD-25010
AWARD	Certificate
ASSESSMENT	430 marks — 258 to pass (60%)
PREREQUISITES	None
VALIDITY	Lifetime
AWARDING BODY	LAPT — London Academy of Professional Training

CURRICULUM

1 Case Studies and Best Practices 0 chapters45 marks	
2 Implementation Challenges 5 chapters · 6 classes65 marks	
• 1 Understanding ISO 25010 Framework and Quality Attributes — 6 classes › 1.1 Explore the ISO 25010 Framework Components › 1.2 Identify Key Quality Attributes in Software Systems › 1.3 Analyze the Importance of Functional Suitability › 1.4 Examine Performance Efficiency in Software Quality › 1.5 Assess Security and Privacy Concerns in Implementation › 1.6 Develop Strategies for Overcoming Quality Attribute Challenges • 2 Identifying Common Implementation Challenges in Software Projects • 3 Analyzing the Impact of Quality Attributes on Implementation Success	• 4 Strategies for Mitigating Implementation Risks • 5 Evaluating Quality Assurance Processes in Software Implementation

• 1 Understanding ISO Standards and Their Importance in Software Quality — 6 classes

- › 1.1 Define ISO Standards and Their Role in Software Quality
- › 1.2 Identify Key Components of ISO 25010
- › 1.3 Explore the Benefits of Implementing ISO 25010
- › 1.4 Analyze Real-World Applications of ISO 25010 in Software Development
- › 1.5 Evaluate the Impact of ISO Compliance on Software Quality Assurance
- › 1.6 Develop an Action Plan for Implementing ISO 25010 in Your Organization

• 2 Overview of ISO 25010 Quality Characteristics — 6 classes

- › 2.1 Define the Key Quality Characteristics of ISO 25010
- › 2.2 Explore Functional Suitability in Software Quality
- › 2.3 Analyze Performance Efficiency as a Quality Attribute
- › 2.4 Discuss Compatibility and its Importance in Systems
- › 2.5 Identify Usability Factors in Software Development
- › 2.6 Evaluate Reliability and Security within ISO 25010 Framework

• 3 Evaluating Software Quality with ISO 25010 Metrics — 6 classes

- › 3.1 Understand the Fundamentals of ISO 25010 Metrics
- › 3.2 Identify Key Quality Characteristics within ISO 25010
- › 3.3 Analyze the Relevance of ISO 25010 Metrics to Software Quality
- › 3.4 Explore Assessment Techniques for ISO 25010 Metrics
- › 3.5 Evaluate Software Products Using ISO 25010 Metrics
- › 3.6 Apply ISO 25010 Metrics for Continuous Quality Improvement

• 4 Implementing ISO 25010 in Software Development Life Cycle — 6 classes

- › 4.1 Understand Key Concepts of ISO 25010
- › 4.2 Identify Quality Characteristics in Software Development
- › 4.3 Analyze the Role of ISO 25010 in the Software Development Life Cycle
- › 4.4 Develop a Framework for Implementing ISO 25010
- › 4.5 Evaluate Tools and Techniques for Compliance with ISO 25010
- › 4.6 Create an Action Plan for ISO 25010 Implementation in Projects

• 5 Case Studies: Successful Application of ISO 25010 — 6 classes

- › 5.1 Explore the Fundamentals of ISO 25010
- › 5.2 Analyze Key Quality Characteristics of ISO 25010
- › 5.3 Examine Real-World Case Studies: Success Stories
- › 5.4 Identify Challenges in Implementing ISO 25010
- › 5.5 Develop Strategies for Effective Application of ISO 25010
- › 5.6 Evaluate the Impact of ISO 25010 on Software Quality

• 1 Foundations of Quality Assurance in Software Development — 6 classes

- › 1.1 Understand the Importance of Quality Assurance in Software Development
- › 1.2 Identify Key Concepts and Terminology in Quality Assurance
- › 1.3 Explore ISO 25010 Quality Characteristics and Their Relevance
- › 1.4 Analyze Common Quality Assurance Strategies in Software Projects
- › 1.5 Evaluate the Role of Leadership in Implementing Quality Assurance Strategies
- › 1.6 Apply Quality Assurance Techniques to Real-World Software Scenarios

• 2 Understanding ISO 25010: Quality Characteristics — 6 classes

- › 2.1 Identify Key Characteristics of ISO 25010
- › 2.2 Analyze Functional Suitability in Quality Models
- › 2.3 Examine Performance Efficiency Metrics
- › 2.4 Assess Usability Criteria for Software Quality
- › 2.5 Evaluate Reliability Attributes in Systems
- › 2.6 Apply ISO 25010 Characteristics in Quality Assurance Strategies

• 3 Implementing Quality Assurance Strategies in Agile Development — 6 classes

- › 3.1 Define Quality Assurance in Agile Development
- › 3.2 Identify Key Quality Assurance Strategies for Agile Teams
- › 3.3 Integrate Continuous Feedback Loops into Agile Processes
- › 3.4 Utilize Automated Testing to Enhance Quality Assurance
- › 3.5 Measure Quality Metrics to Evaluate Agile Performance
- › 3.6 Develop a Quality Assurance Action Plan for Agile Projects

• 4 Automated Testing and Quality Assurance Tools — 6 classes

- › 4.1 Understand Automated Testing Concepts and Benefits
- › 4.2 Identify Different Types of Automated Testing Techniques
- › 4.3 Explore Popular Automated Testing Tools in the Market
- › 4.4 Implement Test Automation Frameworks for Effective Testing
- › 4.5 Evaluate Quality Assurance Strategies Using Automated Testing
- › 4.6 Integrate Automated Testing into Continuous Integration Pipelines

• 5 Evaluating and Measuring Software Quality — 6 classes

- › 5.1 Define Quality Attributes in Software Systems
- › 5.2 Explore the ISO 25010 Quality Model Components
- › 5.3 Analyze Key Metrics for Measuring Software Quality
- › 5.4 Assess Techniques for Evaluating Software Quality
- › 5.5 Implement Quality Assurance Strategies in Software Projects
- › 5.6 Conduct a Case Study on Quality Evaluation Practices

• 1 Understanding ISO 25010: An Overview of Quality Characteristics

— 6 classes

- › 1.1 Identify the Purpose of ISO 25010 in Quality Assurance
- › 1.2 Explore the Key Quality Characteristics Defined in ISO 25010
- › 1.3 Analyze Functional Suitability: Metrics and Examples
- › 1.4 Evaluate Performance Efficiency: Understanding Key Metrics
- › 1.5 Investigate Reliability and Security Metrics within ISO 25010
- › 1.6 Apply ISO 25010 Quality Characteristics to Real-world Scenarios

• 2 Decomposing Quality Characteristics: Functional and Non-Functional Aspects — 6 classes

- › 2.1 Identify Functional Quality Characteristics in Software Systems
- › 2.2 Analyze Non-Functional Quality Characteristics and Their Impact
- › 2.3 Compare Functional and Non-Functional Requirements in Software Development
- › 2.4 Deconstruct Functional Characteristics into Measurable Metrics
- › 2.5 Design a Quality Assessment Framework for Non-Functional Attributes
- › 2.6 Apply Quality Metrics to Evaluate Software Performance and Usability

• 3 Metrics for Assessing Quality: Setting the Standard — 6 classes

- › 3.1 Define Quality Metrics: Understanding Fundamental Concepts
- › 3.2 Identify Key Quality Characteristics: Linking Metrics to Standards
- › 3.3 Explore Common Quality Metrics: A Comparative Analysis
- › 3.4 Analyze Case Studies: Metrics in Real-World Applications
- › 3.5 Develop a Quality Assessment Framework: Practical Implementation
- › 3.6 Evaluate Metrics Effectiveness: Measuring Success and Adjusting Strategies

• 4 Applying Metrics in Real-world Scenarios: Case Studies — 6

classes

- › 4.1 Identify Key Quality Characteristics in Real-world Scenarios
- › 4.2 Analyze Metrics for Assessing Software Usability
- › 4.3 Evaluate Performance Metrics Through Case Studies
- › 4.4 Demonstrate the Use of Reliability Metrics in Software Projects
- › 4.5 Apply Security Metrics to Real-world Applications
- › 4.6 Measure User Satisfaction Using Empirical Case Studies

• 5 Continuous Improvement: Integrating Quality Metrics into the Development Cycle — 6 classes

- › 5.1 Identify Key Quality Metrics in Development Cycle
- › 5.2 Analyze the Role of Continuous Improvement in Quality Assurance
- › 5.3 Integrate Quality Metrics into Agile Development Processes
- › 5.4 Evaluate the Impact of Quality Metrics on Team Performance
- › 5.5 Develop a Continuous Improvement Plan Using Quality Metrics
- › 5.6 Apply Quality Metrics to Monitor Progress and Drive Innovation