LAPT

LONDON ACADEMY OF PROFESSIONAL TRAINING

CERTIFICATE

ISO 29110 — Software Lifecycle Profiles for Very Small Entities

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-29110 • ISO IT & Related Technologies

UKPRN 10067351 • Registered in England and Wales, company 4984798

6	20	102	430	60%	Lifetime
SUBJECTS	CHAPTERS	CLASSES	TOTAL MARKS	PASS MARK	VALIDITY

AT A GLANCE

REFERENCE	**IIT-COD-29110**
AWARD	**Certificate**
ASSESSMENT	**430 marks — 258 to pass (60%)**
PREREQUISITES	**None**
VALIDITY	**Lifetime**
AWARDING BODY	**LAPT — London Academy of Professional Training**

CURRICULUM

1 Best Practices in Software Management

5 chapters · 30 classes · **65 marks**

- **1 Understanding ISO 29110: Framework Overview** — 6 classes
  - 1.1 Define the Purpose of ISO 29110 in Software Management
  - 1.2 Identify Key Components of ISO 29110 Framework
  - 1.3 Explore the Benefits of Implementing ISO 29110
  - 1.4 Analyze Stakeholder Roles within ISO 29110 Framework
  - 1.5 Evaluate Software Lifecycle Profiles for Very Small Entities
  - 1.6 Apply Best Practices from ISO 29110 to Real-World Scenarios
- **2 Defining Software Lifecycle Profiles for Very Small Entities** — 6 classes
  - 2.1 Identify Characteristics of Very Small Entities in Software Development
  - 2.2 Explore Software Lifecycle Approaches Suitable for Small Teams
  - 2.3 Analyze the Importance of Tailoring Software Profiles
  - 2.4 Develop a Basic Software Lifecycle Profile for a Hypothetical Entity
  - 2.5 Evaluate Best Practices in Implementing Lifecycle Profiles
  - 2.6 Create an Action Plan to Adopt ISO 29110 Standards in Your Project
- **3 Implementing Best Practices in Software Development** — 6 classes
  - 3.1 Identify Key Best Practices in Software Development
  - 3.2 Analyze the Impact of Software Development Best Practices
  - 3.3 Develop a Software Development Process Framework
  - 3.4 Implement Effective Communication Strategies in Software Teams
  - 3.5 Evaluate Software Project Risks and Mitigation Techniques
  - 3.6 Create a Continuous Improvement Plan for Software Practices
- **4 Monitoring and Measuring Project Performance** — 6 classes
  - 4.1 Define Key Performance Indicators for Software Projects
  - 4.2 Implement a Monitoring Plan to Track Project Status
  - 4.3 Utilize Metrics to Assess Team Productivity and Efficiency
  - 4.4 Analyze Project Performance Data for Informed Decision Making
  - 4.5 Develop and Adjust Project Plans Based on Performance Insights
  - 4.6 Communicate Performance Results to Stakeholders Effectively
- **5 Continuous Improvement in Software Management** — 6 classes
  - 5.1 Identify Key Metrics for Continuous Improvement in Software Management
  - 5.2 Analyze Current Processes to Uncover Improvement Opportunities
  - 5.3 Implement Agile Methodologies to Foster Adaptability in Software Teams
  - 5.4 Conduct Regular Retrospectives to Review and Revise Practices
  - 5.5 Develop a Culture of Feedback to Enhance Team Collaboration
  - 5.6 Create a Continuous Improvement Plan for Software Lifecycle Management

2 Capstone Project

0 chapters · **65 marks**

3 ISO 29110 Standards Implementation

0 chapters · **65 marks**

4 Leadership and Team Dynamics

5 chapters · 12 classes **65 marks**

- **1 Understanding Leadership Styles in Technology Environments** — 6 classes
  - 1.1 Identify Key Leadership Styles in Technology
  - 1.2 Analyze the Impact of Leadership on Team Dynamics
  - 1.3 Evaluate Situational Leadership in Tech Projects
  - 1.4 Compare Transformational and Transactional Leadership
  - 1.5 Apply Leadership Styles to Real-World Scenarios
  - 1.6 Develop a Leadership Action Plan for Team Improvement
- **2 The Role of Communication in Team Dynamics** — 6 classes
  - 2.1 Understand the Importance of Communication in Teams
  - 2.2 Identify Key Communication Barriers in Team Dynamics
  - 2.3 Explore Effective Verbal and Non-verbal Communication Techniques
  - 2.4 Foster an Open Communication Culture within Teams
  - 2.5 Apply Active Listening Skills for Improved Team Collaboration
  - 2.6 Evaluate Communication Tools and Strategies for Team Success
- **3 Building High-Performing Teams in Software Projects**
- **4 Conflict Resolution and Team Cohesion**
- **5 Adapting Leadership Strategies to Foster Innovation**

5 Process Improvement Techniques

5 chapters · 30 classes **85 marks**

- **1 Understanding Process Improvement in Software Development** — 6 classes
  - 1.1 Define Process Improvement in Software Development
  - 1.2 Identify Key Challenges in Software Development Processes
  - 1.3 Explore Common Process Improvement Models and Techniques
  - 1.4 Assess Current Software Development Practices
  - 1.5 Develop a Strategy for Implementing Process Improvements
  - 1.6 Evaluate the Impact of Process Improvements on Software Quality
- **2 ISO 29110 Framework Overview** — 6 classes
  - 2.1 Understand the Purpose of ISO 29110 Framework
  - 2.2 Identify Key Components of ISO 29110
  - 2.3 Explore the Benefits of Implementing ISO 29110
  - 2.4 Analyze the Software Lifecycle in ISO 29110
  - 2.5 Assess Process Improvement Techniques in Small Entities
  - 2.6 Apply ISO 29110 Principles to Real-World Scenarios
- **3 Identifying Areas for Process Improvement** — 6 classes
  - 3.1 Define Key Metrics for Process Evaluation
  - 3.2 Analyze Current Process Performance Data
  - 3.3 Identify Bottlenecks and Inefficiencies in Workflows
  - 3.4 Gather Stakeholder Feedback for Improvement Insights
  - 3.5 Prioritize Improvement Areas Using Impact Analysis
  - 3.6 Develop Action Plans for Targeted Process Enhancements
- **4 Implementing Process Improvement Techniques** — 6 classes
  - 4.1 Identify Key Areas for Process Improvement
  - 4.2 Analyze Existing Processes Using SWOT Analysis
  - 4.3 Set SMART Goals for Process Improvement Initiatives
  - 4.4 Develop a Step-by-Step Action Plan for Implementation
  - 4.5 Monitor and Measure Improvement Progress Effectively
  - 4.6 Review and Refine Process Improvement Techniques
- **5 Measuring and Sustaining Process Improvements** — 6 classes
  - 5.1 Identify Key Metrics for Measuring Process Improvement
  - 5.2 Implement Data Collection Techniques for Process Assessment
  - 5.3 Analyze Data to Evaluate Process Improvement Outcomes
  - 5.4 Design Feedback Mechanisms to Sustain Improvements
  - 5.5 Develop a Continuous Improvement Plan Based on Metrics
  - 5.6 Communicate Process Improvements to Stakeholders Effectively

6 Software Lifecycle Models

5 chapters · 30 classes **85 marks**

1 Fundamentals of Software Lifecycle Models — 6 classes

- 1.1 Define Software Lifecycle Models and Their Importance
- 1.2 Identify Key Phases in Software Lifecycle Models
- 1.3 Compare Different Software Lifecycle Models
- 1.4 Analyze the Role of Very Small Entities in Software Development
- 1.5 Explore Best Practices for Implementing Software Lifecycle Models
- 1.6 Develop a Mini-Project Using a Software Lifecycle Model

2 Overview of ISO 29110 Framework — 6 classes

- 2.1 Define the Scope and Purpose of ISO 29110
- 2.2 Identify Key Components of the ISO 29110 Framework
- 2.3 Explore the Benefits of Implementing ISO 29110 for Very Small Entities
- 2.4 Compare ISO 29110 with Other Software Lifecycle Models
- 2.5 Analyze Case Studies of Successful ISO 29110 Implementations
- 2.6 Develop a Plan for Adopting ISO 29110 in Your Organization

3 Phases of Software Development Lifecycle — 6 classes

- 3.1 Identify Key Phases in the Software Development Lifecycle
- 3.2 Explore the Planning Phase and Its Importance
- 3.3 Analyze Requirements Gathering Techniques
- 3.4 Examine Design Principles for Effective Software Solutions
- 3.5 Implement Testing Strategies for Software Quality Assurance
- 3.6 Assess the Maintenance Phase and Its Long-Term Benefits

4 Application of Lifecycle Models in Very Small Entities — 6 classes

- 4.1 Understand the Significance of Software Lifecycle Models in Very Small Entities
- 4.2 Identify Key Characteristics of Very Small Entities and their Software Needs
- 4.3 Explore Different Software Lifecycle Models Suitable for Very Small Entities
- 4.4 Analyze the Benefits and Challenges of Implementing Lifecycle Models
- 4.5 Develop a Framework for Selecting an Appropriate Lifecycle Model
- 4.6 Create a Plan for Implementing a Chosen Lifecycle Model in a Very Small Entity

5 Evaluating and Improving Software Lifecycles — 6 classes

- 5.1 Assess Current Software Lifecycle Practices
- 5.2 Identify Key Metrics for Evaluation
- 5.3 Analyze Common Software Lifecycle Models
- 5.4 Compare Effectiveness of Lifecycle Improvements
- 5.5 Implement Best Practices for Lifecycle Optimization
- 5.6 Develop a Continuous Improvement Plan for Software Lifecycles