

ICTQual AB



Qualification Specification

ICTQual AB Level 5 Diploma in Quality Control Software Development



Website
www.ictqualab.co.uk

Email:
support@ictqualab.co.uk

ICTQual AB’s

Level 5 Diploma in Quality Control Software Development

Contents

ICTQual AB Level 5 Diploma in Quality Control Software Development..... 1

About ICTQual AB’s..... 2

Course Overview..... 2

Certification Framework.....4

Entry Requirements.....4

Qualification Structure5

Centre Requirements 2

Support for Candidates 7

Assessment7

Unit Descriptors 8 to 17

Qualification Specification about

ICTQual AB Level 5 Diploma in Quality Control Software Development

About ICTQual AB's

ICTQual AB is a distinguished awarding body based in the United Kingdom, dedicated to fostering excellence in education, training, and skills development. Committed to global standards, ICTQual AB's provides internationally recognized qualifications that empower individuals and organizations to thrive in an increasingly competitive world. Their offerings span diverse industries, including technical fields, health and safety, management, and more, ensuring relevance and adaptability to modern workforce needs.

ICTQual AB's delivers high-quality educational solutions through a network of Approved Training Centres worldwide. Their robust standards and innovative teaching methodologies equip learners with practical knowledge and skills for personal and professional growth. With a mission to inspire lifelong learning and drive positive change, ICTQual AB's continuously evolves its programs to stay ahead of industry trends and technological advancements.

Course Overview

The Level 5 Diploma in Quality Control – Software Development is designed to develop advanced knowledge and practical competencies in applying quality control methodologies throughout the software development lifecycle. This programme provides in-depth understanding of software quality assurance frameworks, testing techniques, auditing methods, process improvement tools, and international quality standards relevant to software development environments. Emphasis is placed on ensuring software reliability, functionality, security, and compliance by integrating quality control strategies into both agile and traditional software development models. Learners are equipped to evaluate code quality, assess risk, implement corrective actions, and lead internal quality assessments to drive continuous improvement in software delivery. This qualification supports learners in mastering the balance between speed of delivery and assurance of quality, ensuring that software systems meet user expectations, regulatory requirements, and organisational goals.

Course Aim

The aim of this qualification is to provide learners with the technical and managerial knowledge necessary to implement, monitor, and improve quality control processes in software development. Learners will be able to apply analytical techniques, implement testing strategies, assess quality metrics, and align software outputs with industry-recognised standards. This course fosters the ability to make informed decisions in complex software environments and promotes the integration of quality throughout the software lifecycle.

For Whom This Course Is For

This qualification is suitable for:

- Quality assurance professionals seeking advancement in the field of software development
- Software developers and testers aiming to deepen their understanding of quality control systems
- IT managers, team leads, and technical coordinators involved in software delivery and performance evaluation
- Individuals transitioning into software quality roles from other quality or engineering disciplines
- Graduates or professionals in IT, computer science, or software engineering who want to gain specialised knowledge in quality control methodologies

Key Standards & Objectives

Upon successful completion of the qualification, learners will be able to:

- Implement quality control frameworks within software development environments
- Apply advanced testing, verification, and validation strategies to ensure code quality
- Interpret and align software practices with international standards such as ISO/IEC 25010 and ISO/IEC 90003
- Analyse defect metrics and quality indicators to improve software processes
- Conduct internal audits and provide quality assurance reporting to support continuous improvement

This programme is structured to meet current industry demands and supports professional advancement in quality assurance within dynamic software settings.

Certification Framework

Qualification title	ICTQual AB Level 5 Diploma in Quality Control Software Development
Course ID	QC0034
Grading Type	Pass / Fail
Competency Evaluation	Coursework / Assignments / Verifiable Experience
Assessment	<p>The assessment and verification process for ICTQual AB's qualifications involves two key stages:</p> <p>Internal Assessment and Verification:</p> <ul style="list-style-type: none">✓ Conducted by the staff at the Approved Training Centre (ATC) to ensure learners meet the required standards through continuous assessments.✓ Internal Quality Assurance (IQA) is carried out by the centre's IQA staff to validate the assessment process. <p>External Quality Assurance:</p> <ul style="list-style-type: none">✓ Managed by ICTQual AB's verifiers, who periodically review the centre's assessment and IQA processes. <p>Verifies that assessments are conducted to the required standards and ensures consistency across centres</p>

Entry Requirements

To enrol in ICTQual AB Level 5 Diploma in Quality Control Software Development, applicants must meet the following entry requirements:

- A **Level 3 qualification** (Certificate or Diploma) in software development, information technology, or quality assurance—or an equivalent qualification in a related field
- A **basic understanding of software development processes**, programming logic, or software testing principles
- Proficiency in **English language** (reading, writing, and comprehension) to engage with course materials and assignments effectively
- **Computer literacy**, including the ability to use web platforms, software applications, and digital tools relevant to quality control
- Learners with relevant **work experience in QA, software testing, or IT support** may also be considered for admission based on prior learning or industry exposure

This qualification is suitable for professionals seeking to advance their software quality assurance expertise, as well as learners looking to gain deeper insights into automation, risk assessment, and testing frameworks within modern development environments.

Qualification Structure

This qualification comprises 10 mandatory units. Candidates must successfully complete all mandatory units to achieve the qualification.

Mandatory Units	
Unit Ref#	Unit Title
QC0034-01	Advanced Software Quality Assurance Principles
QC0034-02	Automation Testing Tools and Frameworks
QC0034-03	Agile Testing and Continuous Integration
QC0034-04	Risk-Based Testing and Quality Metrics
QC0034-05	Performance, Load, and Security Testing
QC0034-06	QA in DevOps Environments
QC0034-07	Test Strategy Design and Optimization
QC0034-08	Software Configuration Management
QC0034-09	Data Integrity and Regulatory Compliance in Software
QC0034-10	Quality Audit and Improvement in Software Projects

Centre Requirements

To ensure quality training delivery, centres must adhere to the following standards:

1. Centre Approval

- ✓ Centres must be formally approved by ICTQual AB's before delivering this qualification.
- ✓ Approval involves a review of facilities, policies, and staff qualifications.

2. Qualified Staff

- ✓ **Tutors:** must possess relevant academic qualifications (minimum Level 6 in Software Development or Quality Assurance) and/or extensive industry experience in software quality control or software testing roles.
- ✓ **Assessors:** Must hold a recognized assessor qualification (e.g., CAVA, AVRA) or equivalent)
- ✓ **Internal Quality Assurers (IQAs):** Must hold a recognized IQA qualification (e.g. Level 4 Award in the IQA and Level 4 Certificate in Leading the IQA) and experience to oversee assessment standards.

3. Learning Facilities

Centre must offer:

- ✓ Private study areas and internet-enabled workspaces (for blended or physical delivery)
- ✓ Academic and pastoral support for learners
- ✓ Administrative support must be available to manage enrolment, tracking, and learner queries efficiently

4. Health and Safety Compliance

- ✓ All training facilities must comply with health and safety regulations.
- ✓ Centres must conduct regular risk assessments for practical activities.

5. Learning Resources

- ✓ **Course Materials:** Approved textbooks, study guides, and digital content must align with the qualification standards.
- ✓ **Assessment Tools:** Templates and guidelines must be provided to ensure standardized evaluation processes.
- ✓ **E-Learning Support:** Centres offering online or blended learning must implement an effective Learning Management System (LMS).

6. Assessment and Quality Assurance

- ✓ Centres must ensure assessments meet ICTQual AB's competency standards.
- ✓ Internal quality assurance (IQA) must be conducted to maintain consistency.
- ✓ External verifiers from ICTQual AB's will review assessment and training practices.

7. Learning Support

- ✓ **Qualification Guidance:** Support for coursework and assignments.
- ✓ **Career Pathway Assistance:** Information on progression opportunities in sustainability and energy sectors.
- ✓ **Accessibility Support:** Accommodations for learners with disabilities or language barriers.

8. Policies and Compliance

Centres must uphold the following policies in accordance with ICTQual AB's standards:

- ✓ Equality, Diversity, and Inclusion Policy.
- ✓ Health and Safety Policy.
- ✓ Safeguarding and Learner Protection Policy.
- ✓ Complaints and Appeals Procedure.
- ✓ Data Protection and Confidentiality Policy.

9. Reporting Requirements

- Centres must provide ICTQual AB's with regular reports on learner registrations, progress, and certification outcomes.

- Assessment records must be maintained for external auditing and quality assurance purposes.

Support for Candidates

Centres should ensure that materials developed to support candidates:

- ✓ Facilitate tracking of achievements as candidate's progress through the learning outcomes and assessment criteria.
- ✓ Include information on how and where ICTQual AB's policies and procedures can be accessed.
- ✓ Provide mechanisms for Internal and External Quality Assurance staff to verify and authenticate evidence effectively.

This approach ensures transparency, supports candidates' learning journeys, and upholds quality assurance standards.

Assessment

This qualification is competence-based, requiring candidates to demonstrate proficiency as defined in the qualification units. The assessment evaluates the candidate's skills, knowledge, and understanding against the set standards. Key details include:

1. Assessment Process:

- ✓ Must be conducted by an experienced and qualified assessor.
- ✓ Candidates compile a portfolio of evidence that satisfies all learning outcomes and assessment criteria for each unit.

2. Types of Evidence:

- ✓ Observation reports by the assessor.
- ✓ Assignments, projects, or reports.
- ✓ Professional discussions.
- ✓ Witness testimonies.
- ✓ Candidate-produced work.
- ✓ Worksheets.
- ✓ Records of oral and written questioning.
- ✓ Recognition of Prior Learning (RPL).

3. Learning Outcomes and Assessment Criteria:

- ✓ **Learning Outcomes:** Define what candidates should know, understand, or accomplish upon completing the unit.
- ✓ **Assessment Criteria:** Detail the standards candidates must meet to demonstrate that the learning outcomes have been achieved.

This framework ensures rigorous and consistent evaluation of candidates' competence in line with the qualification's objectives.

Unit Descriptors

QC0034-01- Advanced Software Quality Assurance Principles

This unit explores the key principles and advanced methods used in software quality assurance (SQA). Learners will understand how to build quality into each stage of the software development lifecycle. The unit covers quality planning, control, and assurance activities in both traditional and agile models. It also explains how to manage quality requirements, prevent defects, and apply standards like ISO/IEC 25010. By the end of the unit, learners will know how to evaluate quality risks, monitor performance, and drive improvement across software teams.

Learning Outcome:	Assessment Criteria:
1. Develop and apply advanced quality assurance methodologies to maintain software reliability, code integrity, and adherence to recognised industry standards.	<div>1.1 Critically evaluate the strengths and limitations of advanced SQA methodologies in high-reliability environments.</div> <div>1.2 Design and implement a multi-layered QA model incorporating static and dynamic testing.</div> <div>1.3 Apply code quality assessment techniques such as cyclomatic complexity and code coverage analysis.</div> <div>1.4 Integrate peer reviews, inspections, and walkthroughs into the QA lifecycle.</div> <div>1.5 Map QA practices to international software engineering standards such as IEEE 730 or ISO/IEC 25010.</div> <div>1.6 Develop a defect taxonomy to classify and prioritise quality issues.</div> <div>1.7 Demonstrate the use of predictive analytics in defect prevention.</div> <div>1.8 Ensure traceability between QA activities and requirements throughout the SDLC.</div> <div>1.9 Implement continuous quality monitoring mechanisms across CI/CD pipelines.</div> <div>1.10 Validate the impact of QA interventions on post-release defect trends.</div>

QC0034-02- Automation Testing Tools and Frameworks

This unit focuses on using automation to improve software testing. Learners will study the main tools, such as Selenium, JUnit, TestNG, and others commonly used in industry. They will learn how to create, manage, and run automated test scripts, and how to design reusable test frameworks. The unit also introduces continuous testing practices and explains how automation fits into agile and DevOps workflows. By completing this unit, learners will be able to select and apply automation tools effectively to increase test speed and accuracy.

Learning Outcome:	Assessment Criteria:
1. Use leading automation tools and frameworks to design, execute, and manage test cases for efficient and scalable testing processes.	<div>1.1 Compare and contrast scripting capabilities of leading automation tools (e.g., Selenium, Cypress, Appium).</div> <div>1.2 Develop and maintain reusable test suites using a selected test automation framework.</div> <div>1.3 Evaluate the scalability of test automation frameworks in cloud-based environments.</div> <div>1.4 Implement test data management strategies in automated testing.</div> <div>1.5 Integrate automated tests with CI/CD tools such as Jenkins, GitLab, or Azure DevOps.</div> <div>1.6 Assess automation ROI and coverage metrics in long-term regression scenarios.</div> <div>1.7 Apply object recognition strategies (XPath, CSS Selectors, AI-based locators) for robust test scripts.</div> <div>1.8 Debug automation failures and maintain automated test code repositories.</div>

QC0034-03- Agile Testing and Continuous Integration

In this unit, learners will understand how testing is done in agile teams. It covers agile values, testing cycles like sprints, and the role of testers in daily stand-ups and sprint reviews. The unit also explains continuous integration (CI) – where code changes are regularly merged and tested. Learners will practise writing user stories, acceptance criteria, and exploratory test cases. They will also understand how to work with tools such as Jenkins or GitLab for CI pipelines. The unit prepares learners to deliver fast, reliable feedback in agile environments.

Learning Outcome:	Assessment Criteria:
1. Integrate agile testing methods within CI pipelines to enable fast, iterative software development and quality validation.	<div>1.1 Apply test-first practices (e.g., TDD, BDD) within agile frameworks.</div> <div>1.2 Coordinate testing with sprint planning, reviews, and retrospectives.</div> <div>1.3 Design lightweight test cases suitable for fast feedback cycles.</div> <div>1.4 Implement automated unit, integration, and acceptance tests in a CI environment.</div> <div>1.5 Evaluate test automation triggers in CI pipelines based on commit types and build stages.</div> <div>1.6 Collaborate with developers and stakeholders to define Definition of Done and Acceptance Criteria.</div> <div>1.7 Assess the impact of test flakiness on agile velocity and delivery predictability.</div> <div>1.8 Perform exploratory testing during continuous delivery cycles.</div> <div>1.9 Monitor and respond to CI test trends to prevent defect accumulation.</div>

QC0034-04- Risk-Based Testing and Quality Metrics

This unit teaches how to plan testing based on risk. Learners will identify areas of software that are more likely to fail or cause serious problems. They will learn how to score risk levels, focus test efforts on critical areas, and report findings using quality metrics. Key concepts such as defect density, test coverage, and mean time to failure are introduced. Learners will also practise making data-driven decisions using risk analysis and test results.

Learning Outcome:	Assessment Criteria:
1. Conduct risk assessments, apply risk-based testing approaches, and utilise quality metrics to drive test planning and decision-making.	<div>1.1 Conduct a comprehensive risk analysis using FMEA or other prioritisation models.</div> <div>1.2 Create a traceability matrix linking risk levels to test coverage.</div> <div>1.3 Determine test priorities based on likelihood, impact, and detectability of software risks.</div> <div>1.4 Apply risk-based partitioning to focus testing efforts on critical components.</div> <div>1.5 Measure and report defect density and distribution against risk exposure.</div> <div>1.6 Evaluate the effectiveness of test coverage using risk-adjusted KPIs.</div> <div>1.7 Design dashboards that visualise quality trends aligned with business risk.</div> <div>1.8 Develop mitigation strategies based on emerging risk profiles.</div> <div>1.9 Incorporate stakeholder risk tolerance into test planning decisions.</div> <div>1.10 Review past project failures to improve future risk-based testing approaches.</div>

QC0034-05- Performance, Load, and Security Testing

This unit introduces non-functional testing methods to check if the software is strong, secure, and reliable under pressure. Learners will explore performance testing, load testing (heavy usage), and stress testing (extreme conditions). They will also learn the basics of security testing – such as checking for weaknesses, unauthorised access, or data leaks. Tools like JMeter, LoadRunner, and OWASP ZAP are discussed. By the end, learners will be able to test systems for speed, stability, and security.

Learning Outcome:	Assessment Criteria:
1. Perform testing focused on system performance, scalability under load, and protection against security vulnerabilities.	<div>1.1 Develop performance test plans aligned with system SLAs.</div> <div>1.2 Simulate realistic user load patterns using tools like JMeter or Gatling.</div> <div>1.3 Analyse server resource utilisation under stress and identify bottlenecks.</div> <div>1.4 Configure test environments that mirror production deployment for accurate load testing.</div> <div>1.5 Perform vulnerability scans using OWASP ZAP or similar tools.</div> <div>1.6 Validate input sanitisation and encryption mechanisms through security testing.</div> <div>1.7 Interpret performance graphs to make scalability decisions.</div> <div>1.8 Execute denial-of-service and authentication robustness tests.</div> <div>1.9 Document performance tuning recommendations based on empirical test data.</div>

QC0034-06- QA in DevOps Environments

This unit explains how quality assurance works in fast-moving DevOps teams. Learners will understand the DevOps culture of shared responsibility between development and operations. They will explore how QA fits into the continuous delivery pipeline and how tools like Docker, Kubernetes, and CI/CD tools support quality. The unit also covers monitoring, real-time feedback, and automated testing within DevOps. Learners will be able to support quality from code creation to software release.

Learning Outcome:	Assessment Criteria:
1. Implement quality assurance within DevOps practices to improve collaboration, deployment cycles, and operational stability.	<div>1.1 Align QA goals with DevOps culture and shared responsibility models.</div> <div>1.2 Automate quality gates within deployment pipelines.</div> <div>1.3 Collaborate with operations teams to implement monitoring and alerting for production quality.</div> <div>1.4 Integrate service-level indicators and objectives (SLIs/SLOs) into QA activities.</div> <div>1.5 Assess deployment readiness based on quality signals from real-time feedback.</div> <div>1.6 Implement canary and blue-green deployment tests.</div> <div>1.7 Use infrastructure-as-code (IaC) practices to validate environment consistency.</div> <div>1.8 Continuously audit test results to refine QA-DevOps alignment.</div>

QC0034-07- Test Strategy Design and Optimization

This unit helps learners design strong and flexible test strategies. It shows how to plan test activities based on business needs, software risks, and user expectations. Learners will study different testing levels (unit, integration, system, acceptance), test types (manual, automated, exploratory), and coverage goals. The unit also focuses on optimising test processes to save time and resources. By the end, learners will know how to create and improve test strategies that meet quality goals.

Learning Outcome:	Assessment Criteria:
1. Design comprehensive test strategies tailored to different software lifecycles, business objectives, and user requirements.	<div>1.1 Define test objectives and entry/exit criteria across software development stages.</div> <div>1.2 Select optimal test design techniques based on application complexity and risk.</div> <div>1.3 Develop modular and scalable test plans aligned with business needs.</div> <div>1.4 Integrate functional, non-functional, and regression strategies into a unified plan.</div> <div>1.5 Balance manual and automated testing in the strategy based on project constraints.</div> <div>1.6 Align testing strategy with customer experience and usability goals.</div> <div>1.7 Incorporate third-party API testing within system integration tests.</div> <div>1.8 Perform cost-benefit analysis of alternative test design options.</div> <div>1.9 Evaluate and update test strategies based on evolving stakeholder expectations.</div> <div>1.10 Benchmark test strategies against industry standards and past project outcomes.</div>

QC0034-08- Software Configuration Management

This unit focuses on keeping software code and related items under control. Learners will understand how to manage code versions, track changes, and control releases. Tools like Git, SVN, or configuration databases are introduced. The unit also explains the importance of change management, build management, and maintaining software integrity. Learners will learn how to avoid version conflicts, ensure proper documentation, and maintain consistency across environments.

Learning Outcome:	Assessment Criteria:
1. Apply version control and configuration management techniques to maintain consistency and traceability in test environments.	<div>1.1 Set up branching strategies (e.g., Gitflow) to manage concurrent test activities.</div> <div>1.2 Tag test artifacts and results with version identifiers for traceability.</div> <div>1.3 Automate build and deployment of test environments using configuration scripts.</div> <div>1.4 Maintain test configuration baselines and change logs.</div> <div>1.5 Integrate test versioning with CI/CD pipelines.</div> <div>1.6 Identify and resolve configuration drift across environments.</div> <div>1.7 Define access control policies for test environment management.</div> <div>1.8 Track dependency changes and their impact on test coverage.</div> <div>1.9 Use containerisation for consistent environment replication.</div>

QC0034-09 Data Integrity and Regulatory Compliance in Software

This unit explains how to ensure data remains accurate, complete, and secure in software systems. Learners will explore good data management practices and how to protect sensitive information. The unit also covers compliance with industry regulations such as GDPR, HIPAA, and software-related standards. Learners will understand the importance of traceability, audit trails, and data security controls in regulated environments. This helps prepare them to support ethical and legal software practices.

Learning Outcome:	Assessment Criteria:
1. Ensure software systems meet data integrity requirements and adhere to regulatory standards through compliant QA processes.	<div>1.1 Validate audit trails, electronic signatures, and change logs for regulatory compliance.</div> <div>1.2 Design test cases to verify data accuracy, completeness, and consistency.</div> <div>1.3 Evaluate software against standards such as 21 CFR Part 11, GDPR, or HIPAA.</div> <div>1.4 Ensure data encryption and access control mechanisms are fully tested.</div> <div>1.5 Document QA evidence to support regulatory inspections and audits.</div> <div>1.6 Monitor data lifecycle from input to archival for integrity assurance.</div> <div>1.7 Implement boundary and validation checks for critical data fields.</div> <div>1.8 Perform periodic compliance audits of test processes and tools.</div>

QC0034-10- Quality Audit and Improvement in Software Projects

This unit introduces internal auditing techniques and continuous improvement in software settings. Learners will study how to plan and carry out audits, assess quality controls, and write audit reports. The unit explains how to identify non-conformities, recommend corrective actions, and follow up on improvement plans. It also covers quality models like PDCA and Six Sigma. Learners will gain the skills to lead quality reviews and help teams improve processes over time.

Learning Outcome:	Assessment Criteria:
1. Conduct internal quality audits, identify gaps, and propose continuous improvement actions for software development projects.	<div>1.1 Develop an internal audit schedule based on project risk and complexity.</div> <div>1.2 Create audit checklists referencing international QA standards.</div> <div>1.3 Perform document reviews, interviews, and system walkthroughs as part of audits.</div> <div>1.4 Identify non-conformities and assess their root causes.</div> <div>1.5 Present audit findings to technical and non-technical stakeholders.</div> <div>1.6 Recommend actionable improvements to test plans and QA practices.</div> <div>1.7 Track implementation of corrective and preventive actions (CAPA).</div> <div>1.8 Assess audit effectiveness through follow-up evaluations.</div> <div>1.9 Integrate audit outcomes into lessons learned databases.</div> <div>1.10 Contribute to organisational process improvement initiatives based on audit trends.</div>

ICTQual AB

Yew Tree Avenue, Dagenham,

London East, United Kingdom RM10 7FN

+447441398083

support@ictqualab.co.uk | www.ictqualab.co.uk

[Visit Official Webpage](http://www.ictqualab.co.uk)

