

ASQ CSQE

ASQ SOFTWARE QUALITY ENGINEER CERTIFICATION QUESTIONS &
ANSWERS

Exam Summary – Syllabus – Questions

CSQE

ASQ Certified Software Quality Engineer (CSQE)

175 Questions Exam – 550/750 Cut Score – Duration of 270 minutes

www.processexam.com

Table of Contents

Know Your CSQE Certification Well:	3
ASQ CSQE Software Quality Engineer Certification Details:	3
CSQE Syllabus:.....	4
I. General Knowledge (16 questions).....	4
II. Software Quality Management (27 questions).....	5
III. System and Software Engineering Processes (32 questions).....	6
IV. Project Management (16 questions)	8
V. Software Metrics and Analysis (21 questions).....	10
VI. Software Verification and Validation (32 questions)	10
VII. Software Configuration Management (16 questions)	12
ASQ CSQE Sample Questions:	13
Study Guide to Crack ASQ Software Quality Engineer CSQE Exam:	16

Know Your CSQE Certification Well:

The CSQE is best suitable for candidates who want to gain knowledge in the ASQ Engineering. Before you start your CSQE preparation you may struggle to get all the crucial Software Quality Engineer materials like CSQE syllabus, sample questions, study guide.

But don't worry the CSQE PDF is here to help you prepare in a stress free manner.

The PDF is a combination of all your queries like-

- What is in the CSQE syllabus?
- How many questions are there in the CSQE exam?
- Which Practice test would help me to pass the CSQE exam at the first attempt?

Passing the CSQE exam makes you ASQ Certified Software Quality Engineer (CSQE). Having the Software Quality Engineer certification opens multiple opportunities for you. You can grab a new job, get a higher salary or simply get recognition within your current organization.

ASQ CSQE Software Quality Engineer Certification Details:

Exam Name	ASQ Certified Software Quality Engineer
Exam Code	CSQE
Exam Fee	ASQ MEMBERS - USD \$433 NON-MEMBERS - USD \$533 RETAKES - USD \$333
Exam Duration	270 Minutes
Number of Questions	175
Passing Score	550/750
Format	Multiple Choice Questions
Books / Trainings	CSQE Handbook
Schedule Exam	Book Your Exam
Sample Questions	ASQ Software Quality Engineer Exam Sample Questions and Answers
Practice Exam	ASQ Certified Software Quality Engineer (CSQE) Practice Test

CSQE Syllabus:

Topic	Details
I. General Knowledge (16 questions)	
A. Benefits of Software Quality Engineering Within the Organization	- Describe the benefits that software quality engineering can have at the organizational level. (Understand)
B. Ethical and Legal Compliance	1. ASQ code of ethics for professional conduct - Determine appropriate behavior in situations requiring ethical decisions such as identifying conflicts of interest and recognizing and resolving ethical issues. (Evaluate) 2. Regulatory and legal issues - Describe the importance of compliance to federal, national, international, and statutory regulations on software development (e.g., EU GDPR and China's Data Protection Laws). Determine the impact of issues such as copyright, trademark, intellectual property rights, product liability, and data privacy [e.g., Health Insurance Portability and Accountability Act (HIPPA) and US Data Protection Laws for personal identifying information (PII)]. (Understand)
C. Standards and Models	- Define and describe the ISO standards such as 9000 and 27000, NIST Special Publication 800-53, IEEE software standards, and the Capability Maturity Model Integration (CMMI) assessment models. (Understand)
D. Leadership Skills	1. Organizational leadership - Use leadership tools and techniques (e.g. organizational change management, knowledge transfer, motivation, mentoring and coaching, and recognition). (Apply) 2. Facilitation skills - Use facilitation and conflict resolution skills and negotiation techniques to manage and resolve issues. Use meeting management tools to maximize meeting effectiveness. (Apply) 3. Communication skills - Identify and apply various communication methods in oral, written, and presentation formats. Apply various techniques for working in multi-cultural environments. Identify and describe the impact that culture, communications, and Diversity, Equity, and Inclusion (DEI) can have on an organizations. (Apply)
E. Team Management	- Use various team management skills including assigning roles and responsibilities, identifying the classic stages of team development (forming, storming, norming, performing, and adjourning), and monitoring and

Topic	Details
	responding to group dynamics. Work with diverse groups and in distributed work environments. Use techniques for working with virtual, in-person, and hybrid teams. (Apply)
II. Software Quality Management (27 questions)	
A. Quality Management System	1. Quality goals and objectives - Design software quality goals and objectives that are consistent with business objectives and regulations. Incorporate software quality goals and objectives into high level program and project plans. Develop and use documents and processes necessary to support software quality management systems. (Create) 2. Quality metrics and monitoring - Determine software quality goals and objectives for measuring and monitoring software including software product and process metrics, and analysis and reporting techniques. (Analyze) 3. Customers and other stakeholders - Assess and evaluate the effect of various stakeholder group requirements (e.g., quality requirements for contractual obligations) on software projects and products. (Evaluate) 4. Outsourcing - Determine the impact that outsourced services can have on organizational goals and objectives. Identify criteria for evaluating suppliers/vendors and subcontractors including governance and quality requirements for contractual obligations. (Analyze) 5. Business continuity, data protection, and data management - Design plans for business continuity, disaster recovery, business documentation, change management, information security, and protection of sensitive and personal data. (Analyze)
B. Methodologies	1. Cost of quality (COQ) and return on investment (ROI) - Analyze cost of quality (COQ) categories (e.g., prevention, appraisal, internal failure, and external failure) and return on investment (ROI) metrics in relation to products and processes. (Analyze) 2. Process improvement - Define and describe elements of benchmarking, lean processes, and the six sigma methodology. Use the Define, Measure, Act, Improve, Control (DMAIC) model and the Plan-Do-Check-Act (PDCA) model for process improvement. (Analyze) 3. Corrective and preventive action procedures

Topic	Details
	- Evaluate corrective and preventive action procedures and impact assessments related to software defects, process nonconformances, and other quality system deficiencies. (Evaluate) 4. Defect prevention - Design and use defect prevention processes such as technical reviews, software tools and technology, and special training. (Evaluate) 5. Metrics and monitoring - Apply metrics and monitoring techniques for adherence to software quality goals and objectives, process improvement, and defect prevention. (Apply)
C. Audits	1. Audit types - Define and distinguish between various audit types (e.g., process, compliance, supplier, system, internal, and external) and methods (e.g., virtual). (Understand) 2. Audit roles and responsibilities - Identify roles and responsibilities for audit participants including the client, lead auditor, audit team members, and auditee. (Understand) 3. Audit process - Define and describe the steps in conducting an audit, developing and delivering an audit report, and determining appropriate follow-up activities. (Apply)
III. System and Software Engineering Processes (32 questions)	
A. Lifecycles and Process Models	1. Waterfall software development lifecycle - Apply the waterfall lifecycle and related process models and identify their benefits and when they are used. (Apply) 2. Incremental / iterative software development lifecycles - Apply the incremental and iterative lifecycles and related process models (e.g., rapid application development methodology) and identify their benefits and when they are used. (Apply) 3. Agile software development lifecycle - Apply the agile lifecycle and related process models (e.g., lean agile and SCRUM methodology) and identify their benefits and when they are used. (Apply) 4. DevOps - Employ DevOps to combine software development and IT operations to continuously release software. (Apply)
B. Systems Architecture	- Identify and describe various architectures (e.g., embedded systems, client-server, n-tier, web, wireless,

Topic	Details
	messaging, and collaboration platforms) and analyze their impact on quality. (Analyze)
C. Cloud computing models and platforms	- Describe cloud computing models such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) and cloud computing platforms such as multi-tenant systems and distributed systems. (Understand)
D. Requirements Engineering	1. Product requirements - Define and describe various types of product and user requirements including but not limited to system, feature, function, non-functional, interface, integration, performance, globalization, and localization. (Understand) 2. Data / information requirements - Define and describe various types of data and information requirements including data management, data integrity, and privacy by design. (Understand) 3. Quality requirements - Define and describe various types of quality requirements, including reliability, and usability. (Understand) 4. Compliance requirements - Define and describe various types of regulatory, safety, and data retention requirements. (Understand) 5. Security requirements - Define and describe various types of security requirements including data security, information security, cybersecurity, data privacy, and encryption requirements for data in transit and at rest. (Understand) 6. Derived requirements - Apply derived requirements such as environmental requirements and user-defined security. (Apply) 7. Non-functional requirements - Define and describe various types of non-functional requirements such as documentation, performance, and technical maintenance. (Understand) 8. Requirements elicitation methods - Describe and use various requirements elicitation methods such as customer needs analysis, use cases, human factors studies, usability prototypes, Joint Application Development (JAD), and storyboards. (Apply) 9. Requirements evaluation - Assess the completeness, consistency, correctness, and testability of requirements and determine their priority. (Evaluate)

Topic	Details
E. Requirements Management	- Requirements change management - Assess the impact that changes to requirements will have on software development processes for all types of life-cycle models. (Evaluate) - Bidirectional traceability - Use various tools and techniques to ensure bidirectional traceability from requirements elicitation and analysis through design and testing. (Apply)
F. Software Analysis, Design, and Development	- Design methods - Identify the steps used in software design and their functions. Define and distinguish between software design methods. (Understand) - Quality attributes and design - Analyze the impact that quality-related elements (e.g., safety, security, reliability, usability, reusability, and maintainability) can have on software design. (Analyze) - Software reuse - Define and distinguish between software reuse, reengineering, and reverse engineering and describe the impact these practices can have on software quality. (Understand) - Software development tools - Analyze and select the appropriate development tools for modeling, code analysis, requirements management, and documentation. (Analyze)
G. Maintenance Management	- Maintenance types - Describe the characteristics of corrective, adaptive, perfective, and preventive maintenance types. (Understand) - Maintenance strategy - Describe various factors affecting the strategy for software maintenance including service-level agreements (SLAs), short- and long-term costs, maintenance releases, and product discontinuance and their impact on software quality. (Understand) - Customer feedback management - Describe the importance of customer feedback management including quality of product support and post-delivery issues analysis and resolution. (Understand)
IV. Project Management (16 questions)	
A. Planning, Scheduling, and Deployment	- Project planning - Use product acceptance, configuring acceptance, forecasts, resources, schedules, task, cost estimates, etc.

Topic	Details
	to develop project plans. (Apply) 2. Work breakdown structure (WBS) - Use work breakdown structure (WBS) in scheduling and monitoring projects. (Apply) 3. Project deployment - Use various tools including milestones, objectives achieved, and task duration to set goals and deploy projects. (Apply)
B. Tracking and Controlling	1. Phase transition control - Use various tools and techniques such as entry/exit criteria, quality gates, Gantt charts, and integrated master schedules to control phase transitions. (Apply) 2. Tracking methods - Calculate project-related costs such as earned value, deliverables, and productivity and track the results against project baselines. (Apply) 3. Project reviews - Use various types of project reviews such as phase-end, management, and retrospectives and post-project reviews to assess project performance and status, to review issues and risks, and to discover and capture lessons learned from the project. (Apply) 4. Program reviews - Define and describe various methods for reviewing and assessing programs in terms of performance, technical accomplishments, and resource utilization. (Understand)
C. Risk Management	1. Risk management methods - Use risk management techniques (e.g. risk impact assessment, prevention, mitigation, and transfer) to evaluate project risks. (Evaluate) 2. Software security risks - Evaluate risks specific to software security including deliberate attacks (e.g., hacking and sabotage), inherent defects that allow unauthorized access to data, and other security breaches. Plan appropriate responses to minimize their impact. Use threat analysis and modeling tools to identify and prevent potential security risks. (Evaluate) 3. Safety and hazard analysis - Evaluate safety risks and hazards related to software development and implementation and determine appropriate steps to minimize their impact. (Evaluate)

Topic	Details
V. Software Metrics and Analysis (21 questions)	
A. Process and Product Measurement	1. Terminology - Define and describe metric and measurement terms such as reliability, internal and external validity, explicit and derived measures, and variation. (Understand) 2. Software product metrics - Choose appropriate metrics to assess various software attributes (e.g., size, complexity, the amount of test coverage needed, requirements volatility, and overall system performance). (Apply) 3. Software process metrics - Measure the effectiveness and efficiency of software processes (e.g., functional verification tests (FVT), cost, yield, customer impact, defect detection, defect containment, total defect containment effectiveness (TDCE), Defect Removal Efficiency (DRE), and process capability). (Apply) 4. Data integrity - Describe the importance of data integrity from planning through collection and analysis. Apply various techniques to ensure data quality, accuracy, completeness, and timeliness. (Apply)
B. Analysis and Reporting Techniques	1. Metric reporting tools - Using various metric representation tools such as dashboards and stoplight charts to report results. (Apply) 2. Classic quality tools - Describe the appropriate use of classic quality tools (e.g., flowcharts, Pareto charts, cause and effect diagrams, control charts, and histograms). (Apply) 3. Problem-solving tools - Describe the appropriate use of problem-solving tools (e.g., affinity, tree, matrix, activity network, root cause analysis, and Data Flow Diagrams (DFD)). (Apply)
VI. Software Verification and Validation (32 questions)	
A. Theory	1. V&V methods - Use software verification and validation methods (e.g., static analysis, structural analysis, mathematical proof, simulation, and automation) and determine which tasks should be iterated because of modifications. (Apply) 2. Software product evaluation - Use various evaluation methods on documentation, source code, etc. to determine whether user needs and

Topic	Details
	project objectives have been satisfied. (Analyze)
B. Test Planning and Design	1. Test strategies - Select and analyze test strategies (e.g., test-driven design, good-enough, riskbased, time-box, top-down, bottom-up, black-box, white-box, simulation, automation, and continuous) for various situations such as Software as a Service (SaaS) and new product applications. (Analyze) 2. Test plans - Develop and evaluate test plans and procedures such as system, acceptance, and validation to determine whether project objectives are being met and risks are appropriately mitigated. (Create) 3. Test designs - Select and evaluate various test designs, including fault insertion, fault-error handling, equivalence class partitioning, boundary value. (Evaluate) 4. Software tests - Evaluate and execute various tests including unit, functional, performance, integration, regression, usability, acceptance, certification, environmental load, stress, worst-case, perfective, exploratory, and system. (Evaluate) 5. Tests of external products - Determine appropriate levels of testing for integrating supplier, third-party, and subcontractor components and products. (Apply) 6. Test coverage specifications - Evaluate the adequacy of test specifications such as functions, states, data and time domains, interfaces, security, and configurations that include internationalization and platform variances. (Evaluate) 7. Code coverage techniques - Use and identify various tools and techniques to facilitate code coverage analysis techniques such as branch coverage, condition, domain, and boundary. (Apply) 8. Test environments - Select and use simulations, test libraries, drivers, stubs, harnesses, etc. and identify parameters to establish a controlled test environment. (Analyze) 9. Test tools - Identify and use test utilities, diagnostics, automation, and test management tools. (Apply) 10. Test data management - Ensure the integrity and security of test data through the

Topic	Details
	use of configuration controls. (Apply)
C. Reviews and Inspections	- Use desk-checks, peer reviews, walk-throughs, inspections, etc. to identify defects. (Apply)
D. Test Execution Documents	- Evaluate and manage test execution documents such as test results, defect reporting and tracking records, test completion metrics, trouble reports, and input/output specifications. (Evaluate)
VII. Software Configuration Management (16 questions)	
A. Configuration Infrastructure	1. Configuration management team - Describe the roles and responsibilities of a configuration management group. (Understand) [NOTE: The roles and responsibilities of the configuration control board (CCB) are covered in area VII.C.2.] 2. Configuration management tools - Describe configuration management tools as they are used for managing libraries, build systems, and defect tracking systems. (Understand) 3. Library processes - Describe dynamic, static, and controlled library processes and related procedures such as check-in/check-out and merge changes. (Understand)
B. Configuration Identification	1. Configuration items - Describe software configuration items (e.g., baselines, documentation, software code, and equipment) and identification methods (e.g., naming conventions and versioning schemes). (Understand) 2. Software builds and baselines - Describe the relationship between software builds and baselines and describe methods for controlling builds and baselines (e.g., automation and new versions). (Understand)
C. Configuration Control and Status Accounting	1. Item change and version control - Describe processes for documentation control, item change tracking, and version control that are used to manage various configurations. Describe processes used to manage configuration item dependencies in software builds and versioning. (Understand) 2. Configuration control board (CCB) - Describe the roles, responsibilities and processes of the Configuration Control Board (CCB). (Understand) [NOTE: The roles and responsibilities of the configuration

Topic	Details
	management team are covered in area VII.A.1.] 3. Concurrent development - Describe the use of configuration management control principles in concurrent development processes. (Understand) 4. Status accounting - Apply various processes for establishing, maintaining, and reporting the status of configuration items such as baselines, builds, and tools.(Apply)
D. Configuration Audits	- Define and distinguish between functional and physical configuration audits and how they are used in relation to product specification. (Understand)
E. Product Release and Distribution	1. Product release - Assess the effectiveness of product release processes: planning, scheduling, and defining hardware and software dependencies. (Evaluate) 2. Customer deliverables - Assess the completeness of customer deliverables including packaged, hosted, and downloadable products; license keys and user documentation; and marketing and training materials. (Evaluate) 3. Archival processes - Assess the effectiveness of source and release archival processes: backup planning and scheduling, data retrieval, archival of build environments, retention of historical records, and offsite storage. (Evaluate)

ASQ CSQE Sample Questions:

Question: 1

When evaluating risks related to software security, what should be included in the response plan to minimize their impact? (Choose Three)

- a) Incident response procedures
- b) Regular system backups
- c) Clear communication channels
- d) Employee satisfaction surveys

Answer: a, b, c

Question: 2

What are key facilitation skills for managing and resolving conflicts? (Choose Two)

- a) Dictating solutions
- b) Using negotiation techniques
- c) Ignoring the issue
- d) Employing conflict resolution skills

Answer: b, d

Question: 3

Which of the following are benefits of implementing software quality engineering at the organizational level? (Choose Two)

- a) Improved product reliability
- b) Increased market competition
- c) Enhanced customer satisfaction
- d) Higher development costs

Answer: a, c

Question: 4

How does reverse engineering differ from software reengineering?

- a) Reverse engineering involves creating new software components
- b) Reverse engineering aims to understand and analyze existing software
- c) Software reengineering focuses on understanding the existing software's structure
- d) Software reengineering is used for hardware design

Answer: b

Question: 5

When evaluating requirements, which aspects are essential to assess? (Choose Three)

- a) Completeness?
- b) Consistency?
- c) Aesthetics?
- d) Testability?

Answer: a, b, d

Question: 6

What is the main goal of conducting code reviews and inspections?

- a) To slow down the development process
- b) To identify defects early
- c) To train new developers
- d) To prepare for deployment

Answer: b

Question: 7

In the context of Six Sigma, what does the "Define" phase focus on?

- a) Collecting data on current processes
- b) Implementing solutions to process problems
- c) Identifying and defining the problem
- d) Monitoring the process improvements

Answer: c

Question: 8

Return on Investment (ROI) is an important metric in software quality management. Which of the following best describes ROI?

- a) The ratio of net profit to the total investment cost
- b) The cost associated with identifying defects in the product
- c) The costs resulting from delivering defective products to customers
- d) The costs incurred to train employees

Answer: a

Question: 9

Which skills are essential for managing virtual, in-person, and hybrid teams? (Choose Two)

- a) Clear communication
- b) Micromanagement
- c) Flexibility in scheduling
- d) Centralized decision-making

Answer: a, c

Question: 10

What is the main goal of conducting code reviews and inspections?

- a) To slow down the development process
- b) To identify defects early
- c) To train new developers
- d) To prepare for deployment

Answer: b

Study Guide to Crack ASQ Software Quality Engineer CSQE Exam:

- Getting details of the CSQE syllabus, is the first step of a study plan. This pdf is going to be of ultimate help. Completion of the syllabus is must to pass the CSQE exam.
- Making a schedule is vital. A structured method of preparation leads to success. A candidate must plan his schedule and follow it rigorously to attain success.
- Joining the ASQ provided training for CSQE exam could be of much help. If there is specific training for the exam, you can discover it from the link above.
- Read from the CSQE sample questions to gain your idea about the actual exam questions. In this PDF useful sample questions are provided to make your exam preparation easy.
- Practicing on CSQE practice tests is must. Continuous practice will make you an expert in all syllabus areas.

Reliable Online Practice Test for CSQE Certification

MakeProcessExam.com your best friend during your ASQ Certified Software Quality Engineer exam preparation. We provide authentic practice tests for the CSQE exam. Experts design these online practice tests, so we can offer you an exclusive experience of taking the actual CSQE exam. We guarantee you 100% success in your first exam attempt if you continue practicing regularly. Don't bother if you don't get 100% marks in initial practice exam attempts. Just utilize the result section to know your strengths and weaknesses and prepare according to that until you get 100% with our practice tests. Our evaluation makes you confident, and you can score high in the CSQE exam.

Start Online Practice of CSQE Exam by Visiting URL

<https://www.processexam.com/asq/asq-certified-software-quality-engineer-csqe>