Certified Software Quality Engineer (CSQE)
Body of Knowledge Map 2016 - 2023

The Certified Software Quality Engineer (CSQE) Body of Knowledge (BoK) has been updated to ensure that the most current state of CSQE practice is being tested in the examination. If you would like more information on how a BoK is updated, see a description of the process on https://asq.org/cert/exam-development.

The first step in updating the CSQE BoK was conducting a job analysis survey to determine whether the topics in the 2016 BoK are still relevant to the job role of CSQEs and to identify any new topics that have emerged since that BoK was developed. The results of the CSQE job analysis survey showed that all except two topics that were in the previous BOK are still relevant to the job roles of software quality engineers currently. However, several new topics were added to the previous BOK because of changing industry needs. As indicated in Table 2: one new topic was added (III.C), four new subtopics were added (II.A.2, II.B.5, III.A.4, III.C.6), two subtopics were combined into one topic (I.E.), and several subtexts were revised.

The 2023 CSQE Body of Knowledge will be introduced at the **December 2023** administration. Both BoKs will be available online until February 1, 2024, at which time the 2016 BoK will be removed.

General comments about ASQ Body of Knowledge updates

When the BoK is updated for an ASQ exam, most of the material covered in the BoK remains the same. There are very few programs that change significantly over 5-7 years. ASQ informs all the exam development committees that ASQ Certification Exams must reflect “the state of practice” not “the state of the art.” This helps to keep the programs grounded in what people currently do rather than being driven by the latest hot-topic improvement idea or trend. Typically, the biggest change in any updated BoK is in how the content is organized. When a new BoK is announced and posted on the ASQ website, we also include a “BoK Map” that highlights the changes between the two BoKs: old and new. The BoK Map also clearly identifies any new content that has been added to the exam and any content that has been removed from the exam.

Regarding exam preparation materials, you should be able to use any of the reference books that are currently listed on the bibliography for the exam program. These are the source materials that the exam development committees use to write questions and verify answers.

Specific comments about the 2023 CSQE Body of Knowledge updates

The CSQE Body of Knowledge mostly stayed the same with the 2023 update. In Section I, two subtopics were combined into one topic (I.E.) In Section II, two new subtopics were added: Quality Metrics and Monitoring (II.A.2) and Metrics and Monitoring (II.B.5). In Section III, one new topic was added: Cloud Computing Models and Platforms (III.C), and three subtopics were added to Section III: DevOps (III.A.4), Derived Requirements (III.D.6), and Non-functional Requirements (III.D.7). There were no major changes to content in Section IV, Section V, Section VI, and Section VII. There were several revisions made to the subtext including adding new pieces of knowledge. There were four subtopics that increased in level of cognitive: II.A.2, II.B.2, VI.B.4, and VII.C.4. For each of these topics, the subtext was revised to reflect the new cognitive level.

Table 1 (below) portrays the change in items allocated to each section of the BoK. The section names have all remained the same. Table 2 (on page 3) presents the 2023 CSQE BoK and maps the topics to the 2016 BoK. Table 3 (page 11) presents the 2016 CSQE BoK and maps the topics to the 2023 BoK. Details on changes between the two can be found below.

Table 1. CSQE BoK Section Item Allocation

BoK Section	**2016 BoK**	**2023 BoK**	**Difference**
I. General Knowledge	16	16	**0**
II. Software Quality Management	22	27	**+5**
III. System and Software Engineering Processes	32	32	**0**
IV. Project Management	22	16	**-6**
V. Software Metrics and Analysis	19	21	**+2**
VI. Software Verification and Validation	29	32	**+3**
VII. Software Configuration Management	20	16	**-4**

Table 2. 2023 CSQE BoK mapped to 2016 CSQE BoK

2016 BoK	2023 BoK Details	Notes
	I. General Knowledge (16 questions)	
I.A	**A. Benefits of software quality engineering within the organization** Describe the benefits that software quality engineering can have at the organizational level. (Understand)	
I.B	**B. Ethical and legal compliance**	
I.B.1	**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)	
I.B.2	**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)	Added international regulations. Added EU GDPR and China's Data Protection Laws as examples. Added trademark. Added Health Insurance Portability and Accountability act (HIPPA) and US Data Protection Laws for personal identifying information (PII) as examples.
I.C	**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 SEI Capability Maturity Model Integration (CMMI) for development, services, and acquisition assessment models. (Understand)	Added ISO 27000 and NIST Special Publication 800-53.

I.D	**D. Leadership skills**	
I.D.1	**1. Organizational leadership** Use leadership tools and techniques (e.g. organizational change management, knowledge transfer, motivation, mentoring and coaching, and recognition). (Apply)	
I.D.2	**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)	
I.D.3	**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 organization. (Apply)	Added Diversity, Equity, and Inclusion (DEI).
I.E.1 & I.E.2	**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 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)	Combined subtopics “team management” and “team tools” into one topic. Changed the name of the topic from “Team Skills” to “Team Management”
II	**II. Software Quality Management (27 questions)**	Added 5 questions
II.A	**A. Quality management system**	

II.A.1	**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)	Added regulations.
NEW!	**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)	NEW! subtopic.
II.A.2	**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)	Increased the cognitive level from Analyze to Evaluate. Modified subtext to reflect the “Evaluate” cognitive level. Added quality requirements for contractual obligations as an example.
II.A.3	**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)	Added governance and quality requirements for contractual obligations.
II.A.4	**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)	
II.B	**B. Methodologies**	
II.B.1	**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)	

II.B.2	**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)	Increased the cognitive level from Apply to Analyze. Modified the subtext language to reflect the “Analyze” cognitive level.
II.B.3	**3. Corrective and preventive action procedures** Evaluate corrective and preventive action procedures and impact assessments related to software defects, process nonconformances, and other quality system deficiencies. (Evaluate)	Added preventive to the subtopic title. Added preventive action and impact assessments.
II.B.4	**4. Defect prevention** Design and use defect prevention processes such as technical reviews, software tools and technology, and special training. (Evaluate)	
NEW!	**5. Metrics and monitoring** Apply metrics and monitoring techniques for adherence to software quality goals and objectives, process improvement, and defect prevention. (Apply)	NEW! subtopic.
II.C	**C. Audits**	
II.C.1	**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)	Added internal and external audits as examples. Added methods.
II.C.2	**2. Audit roles and responsibilities** Identify roles and responsibilities for audit participants including the client, lead auditor, audit team members, and auditee. (Understand)	
II.C.3	**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	**III. System and Software Engineering Processes (32 questions)**	

III.A	**A. Lifecycles and Process Models**	
III.A.1	**1. Waterfall software development lifecycle** Apply the waterfall lifecycle and related process models and identify their benefits and when they are used. (Apply)	
III.A.2	**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)	Added rapid application development methodology as an example.
III.A.3	**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)	Added lean agile and SCRUM methodology as examples.
NEW!	**4. DevOps** Employ DevOps to combine software development and IT operations to continuously release software. (Apply)	NEW! subtopic.
III.B	**B. Systems architecture** Identify and describe various architectures (e.g., embedded systems, client-server, n-tier, web, wireless, messaging, and collaboration platforms) and analyze their impact on quality. (Analyze)	
NEW!	**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 (IaS) and cloud computing platforms such as multi-tenant systems and distributed systems. (Understand)	NEW! topic.
III.C	**D. Requirements engineering**	
III.C.1	**1. Product requirements** Define and describe various types of product and user requirements including but not limited to system, feature, function, interface, integration, performance, globalization, and localization. (Understand)	Added user requirements.
III.C.2	**2. Data / information requirements** Define and describe various types of data and information requirements including data management, data integrity, and privacy by design. (Understand)	Added privacy by design.
III.C.3	**3. Quality requirements** Define and describe various types of quality requirements, including reliability, usability. (Understand)	

III.C.4	**4. Compliance requirements** Define and describe various types of regulatory, safety, and data retention requirements. (Understand)	Added data retention.
III.C.5	**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)	Added encryption requirements for data in transit and at rest.
NEW!	**6. Derived requirements** Apply derived requirements such as environmental requirements and user-defined security. (Apply)	NEW! subtopic.
NEW!	**7. Derived requirements** Define and describe various types of non-functional requirements such as documentation, performance, and technical maintenance. (Understand)	NEW! subtopic.
III.C.6	**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)	
III.C.7	**9. Requirements evaluation** Assess the completeness, consistency, correctness, and testability of requirements and determine their priority. (Evaluate)	
III.D	**E. Requirements management**	
III.D.1	**1. Requirements change management** Assess the impact that changes to requirements will have on software development processes for all types of life-cycle models. (Evaluate)	
III.D.2	**2. Bidirectional traceability** Use various tools and techniques to ensure bidirectional traceability from requirements elicitation and analysis through design and testing. (Apply)	
III.E	**F. Software analysis, design, and development**	
III.E.1	**1. Design methods** Identify the steps used in software design and their functions. Define and distinguish between software design methods. (Understand)	

III.E.2	**2. 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)	
III.E.3	**3. Software reuse** Define and distinguish between software reuse, reengineering, and reverse engineering and describe the impact these practices can have on software quality. (Understand)	
III.E.4	**4. Software development tools** Analyze and select the appropriate development tools for modeling, code analysis, requirements management, and documentation. (Analyze)	
III.F	**G. Maintenance management**	
III.F.1	**1. Maintenance types** Describe the characteristics of corrective, adaptive, perfective, and preventive maintenance types. (Understand)	
III.F.2	**2. 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)	
III.F.3	**3. Customer feedback management** Describe the importance of customer feedback management including quality of product support and post-delivery issues analysis and resolution. (Understand)	
IV.	**IV. Project Management (16 questions)**	Subtracted 6 questions
IV.A	**A. Planning, scheduling, and deployment**	
IV.A.1	**1. Project planning** Use product acceptance, configurating acceptance, forecasts, resources, schedules, task, cost estimates, etc. to develop project plans. (Apply)	Added product acceptance and configurating acceptance.
IV.A.2	**2. Work Breakdown Structure (WBS)** Use Work Breakdown Structure (WBS) in scheduling and monitoring projects. (Apply)	

IV.A.3	**3. Project deployment** Use various tools including milestones, objectives achieved, and task duration to set goals and deploy projects. (Apply)	
IV.B	**B. Tracking and controlling**	
IV.B.1	**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)	
IV.B.2	**2. Tracking methods** Calculate project-related costs such as earned value, deliverables, and productivity and track the results against project baselines. (Apply)	
IV.B.3	**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)	
IV.B.4	**4. Program reviews** Define and describe various methods for reviewing and assessing programs in terms of performance, technical accomplishments, and resource utilization. (Understand)	
IV.C	**C. Risk management**	
IV.C.1	**1. Risk management methods** Use risk management techniques (e.g., risk impact assessment, prevention, mitigation, and transfer) to evaluate project risks. (Evaluate)	Added risk impact assessment as an example.
IV.C.2	**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)	Added threat analysis and modeling tools to identify and prevent potential security risks.
IV.C.3	**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)	

V	**V. Software Metrics and Analysis (21 questions)**	Added 2 questions
V.A	**A. Process and product measurement**	
V.A.1	**1. Terminology** Define and describe metric and measurement terms such as reliability, internal and external validity, explicit and derived measures, and variation. (Understand)	
V.A.2	**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)	Added requirements volatility.
V.A.3	**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)	
V.A.4	**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)	
V.B	**B. Analysis and reporting techniques**	
V.B.1	**1. Metric reporting tools** Using various metric representation tools such as dashboards and stoplight charts to report results. (Apply)	
V.B.2	**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)	
V.B.3	**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	**VI. Software Verification and Validation (32 questions)**	Added 3 questions

VI.A	**A. Theory**	
VI.A.1	**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)	
VI.A.2	**2. Software product evaluation** Use various evaluation methods on documentation, source code, etc. to determine whether user needs and project objectives have been satisfied. (Analyze)	
VI.B	**B. Test planning and design**	
VI.B.1	**1. Test strategies** Select and analyze test strategies (e.g., test-driven design, good-enough, risk-based, 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)	Added Software as a Service (SaaS) and new product applications as examples.

VI.B.2	**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)	
VI.B.3	**3. Test designs** Select and evaluate various test designs, including fault insertion, fault-error handling, equivalence class partitioning, boundary value. (Evaluate)	
VI.B.4	**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)	Increased the cognitive level from Apply to Evaluate. Modified the subtext language to reflect the “Evaluate” cognitive level.
VI.B.5	**5. Tests of external products** Determine appropriate levels of testing for integrating supplier, third-party, and subcontractor components and products. (Apply)	
VI.B.6	**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)	
VI.B.7	**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)	
VI.B.8	**8. Test environments** Select and use simulations, test libraries, drivers, stubs, harnesses, etc. and identify parameters to establish a controlled test environment. (Analyze)	
VI.B.9	**9. Test tools** Identify and use test utilities, diagnostics, automation, and test management tools. (Apply)	
VI.B.10	**10. Test data management** Ensure the integrity and security of test data using configuration controls. (Apply)	

VI.C	**C. Reviews and inspections** Use desk-checks, peer reviews, walk-throughs, inspections, etc. to identify defects. (Apply)	
VI.D	**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)	Added manage.
VII	**VII. Software Configuration Management (16 questions)**	Subtracted 4 questions
VII.A	**A. Configuration infrastructure**	
VII.A.1	**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.]	
VII.A.2	**2. Configuration management tools** Describe configuration management tools as they are used for managing libraries, build systems, and defect tracking systems. (Understand)	
VII.A.3	**3. Library processes** Describe dynamic, static, and controlled library processes and related procedures such as check-in/check-out and merge changes. (Understand)	
VII.B	**B. Configuration identification**	
VII.B.1	**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)	
VII.B.2	**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)	
VII.C	**C. Configuration control and status accounting**	

VII.C.1	**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)	
VII.C.2	**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 management team are covered in area VII.A.1.]	
VII.C.3	**3. Concurrent development** Describe the use of configuration management control principles in concurrent development processes. (Understand)	
VII.C.4	**4. Status accounting** Apply various processes for establishing, maintaining, and reporting the status of configuration items such as baselines, builds, and tools. (Apply)	Increased the cognitive level from Understand to Apply. Modified the subtext language to reflect the "Apply" cognitive level.
VII.D	**D. Configuration audits** Define and distinguish between functional and physical configuration audits and how they are used in relation to product specification. (Understand)	
VII.E	**E. Product release and distribution**	
VII.E.1	**1. Product release** Assess the effectiveness of product release processes: planning, scheduling, and defining hardware and software dependencies. (Evaluate)	
VII.E.2	**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)	
VII.E.3	**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)	

Table 3. 2016 CSQE BoK mapped to the 2023 CSQE BoK

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
I	General Knowledge	**I**	General Knowledge	
I.A.1	Benefits of software quality engineering within the organization	**I.A.1**	Benefits of software quality engineering within the organization	
I.B	Ethical and legal compliance	**I.B**	Ethical and legal compliance	
I.B.1	ASQ Code of Ethics for professional conduct	**I.B.1**	ASQ Code of Ethics for professional conduct	
I.B.2	Regulatory and legal issues	**I.B.2**	Regulatory and legal issues	Added international regulations. Added EU GDPR and China's Data Protection Laws as examples. Added trademark. Added Health Insurance Portability and Accountability act (HIPPA) and US Data Protection Laws for personal identifying information (PII) as examples.
I.C	Standards and models	**I.C**	Standards and models	Added ISO 27000 and NIST Special Publication 800-53.
I.D	Leadership skills	**I.D**	Leadership skills	
I.D.1	Organizational leadership	**I.D.1**	Organizational leadership	
I.D.2	Facilitation skills	**I.D.2**	Facilitation skills	
I.D.3	Communication skills	**I.D.3**	Communication skills	Added Diversity, Equity, and Inclusion (DEI).
I.E	Team skills	**I.E**	Team management	Combined subtopics "team management" and "team tools" into one topic. Changed the name of the topic from "Team Skills" to "Team Management"

2016 BoK		2023 BoK		Notes
Code	Label	Code	Label	
I.E.1	Team management			
I.E.2	Team tools			

2016 BoK		2023 BoK		Notes
Code	Label	Code	Label	
II	Software Quality Management	**II**	Software Quality Management	Added 5 questions
II.A	Quality management system	**II.A**	Quality management system	
II.A.1	Quality goals and objectives	**II.A.1**	Quality goals and objectives	**Added regulations.**
		II.A.2	Quality metrics and monitoring	NEW! subtopic.
II.A.2	Customers and other stakeholders	**II.A.3**	Customers and other stakeholders	Increased the cognitive level from Analyze to Evaluate. Modified subtext to reflect the “Evaluate” cognitive level. Added quality requirements for contractual obligations as an example.
II.A.3	Outsourcing	**II.A.4**	Outsourcing	Added governance and quality requirements for contractual obligations.
II.A.4	Business continuity, data protection, and data management	**II.A.5**	Business continuity, data protection, and data management	
II.B	Methodologies	**II.B**	Methodologies	
II.B.1	Cost of Quality (COQ) and Return on Investment (ROI)	**II.B.1**	Cost of Quality (COQ) and Return on Investment (ROI)	
II.B.2	Process improvement	**II.B.2**	Process improvement	Increased the cognitive level from Apply to Analyze. Modified the subtext language to reflect the “Analyze” cognitive level.
II.B.3	Corrective action procedures	**II.B.3**	Corrective and preventive action procedures	Added preventive to the subtopic title. Added preventive action and impact assessments.
II.B.4	Defect prevention	**II.B.4**	Defect prevention	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
		II.B.5	Metrics and monitoring	NEW! subtopic.
II.C	Audits	**II.C**	Audits	
II.C.1	Audit types	**II.C.1**	Audit types	Added internal and external audits as examples. Added methods.
II.C.2	Audit roles and responsibilities	**II.C.2**	Audit roles and responsibilities	
II.C.3	Audit process	**II.C.3**	Audit process	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
III	System and Software Engineering Processes	**III**	System and Software Engineering Processes	
III.A	Lifecycles and process models	**III.A**	Lifecycles and process models	
III.A.1	Waterfall software development lifecycle	**III.A.1**	Waterfall software development lifecycle	Added rapid application development methodology as an example.
III.A.2	Incremental / iterative software development lifecycles	**III.A.2**	Incremental / iterative software development lifecycles	Added lean agile and SCRUM methodology as examples.
III.A.3	Agile software development lifecycle	**III.A.3**	Agile software development lifecycle	NEW! subtopic.
		III.A.4	DevOps	
III.B	Systems architecture	**III.B**	Systems architecture	NEW! topic.
		III.C	Cloud computing models and platforms	
III.C	Requirements engineering	**III.D**	Requirements engineering	Added user requirements.
III.C.1	Product requirements	**III.D.1**	Product requirements	Added privacy by design.
III.C.2	Data / information requirements	**III.D.2**	Data / information requirements	
III.C.3	Quality requirements	**III.D.3**	Quality requirements	Added data retention.
III.C.4	Compliance requirements	**III.D.4**	Compliance requirements	Added encryption requirements for data in transit and at rest.
III.C.5	Security requirements	**III.D.5**	Security requirements	
NEW!		**III.D.6**	Derived requirements	NEW! subtopic.
NEW!		**III.D.7**	Non-functional requirements	NEW! subtopic.
III.C.6	Requirements elicitation methods	**III.D.8**	Requirements elicitation methods	
III.C.7	Requirements evaluation	**III.D.9**	Requirements evaluation	
III.D	Requirements management	**III.E**	Requirements management	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
III.D.1	Requirements change management	**III.E.1**	Requirements change management	
III.D.2	Bidirectional traceability	**III.E.2**	Bidirectional traceability	
III.E	Software analysis, design, and development	**III.F**	Software analysis, design, and development	
III.E.1	Design methods	**III.F.1**	Design methods	
III.E.2	Quality attributes and design	**III.F.2**	Quality attributes and design	
III.E.3	Software reuse	**III.F.3**	Software reuse	
III.E.4	Software development tools	**III.F.4**	Software development tools	
III.F	Maintenance management	**III.G**	Maintenance management	
III.F.1	Maintenance types	**III.G.1**	Maintenance types	
III.F.2	Maintenance strategy	**III.G.2**	Maintenance strategy	
III.F.3	Customer feedback management	**III.G.3**	Customer feedback management	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
IV.	Project Management	**IV.**	Project Management	Subtracted 6 questions
IV.A	Planning, scheduling, and deployment	**IV.A**	Planning, scheduling, and deployment	
IV.A.1	Project planning	**IV.A.1**	Project planning	Added product acceptance and configurating acceptance.
IV.A.2	Work Breakdown Sructure (WBS)	**IV.A.2**	Work Breakdown Structure (WBS)	
IV.A.3	Project deployment	**IV.A.3**	Project deployment	
IV.B	Tracking and controlling	**IV.B**	Tracking and controlling	
IV.B.1	Phase transition control	**IV.B.1**	Phase transition control	
IV.B.2	Tracking methods	**IV.B.2**	Tracking methods	
IV.B.3	Project reviews	**IV.B.3**	Project reviews	
IV.B.4	Program reviews	**IV.B.4**	Program reviews	
IV.C	Risk management	**IV.C**	Risk management	
IV.C.1	Risk management methods	**IV.C.1**	Risk management methods	Added risk impact assessment as an example.
IV.C.2	Software security risks	**IV.C.2**	Software security risks	Added threat analysis and modeling tools to identify and prevent potential security risks.
IV.C.3	Safety and hazard analysis	**IV.C.3**	Safety and hazard analysis	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
V	Software Metrics and Analysis	**V**	Software Metrics and Analysis	Added 2 questions
V.A	Process and product measurement	**V.A**	Process and product measurement	
V.A.1	Terminology	**V.A.1**	Terminology	
V.A.2	Software product metrics	**V.A.2**	Software product metrics	Added requirements volatility.
V.A.3	Software process metrics	**V.A.3**	Software process metrics	
V.A.4	Data integrity	**V.A.4**	Data integrity	
V.B	Analysis and reporting techniques	**V.B**	Analysis and reporting techniques	
V.B.1	Metric reporting tools	**V.B.1**	Metric reporting tools	
V.B.2	Classic quality tools	**V.B.2**	Classic quality tools	
V.B.3	Problem-solving tools	**V.B.3**	Problem-solving tools	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
VI	Software Verification and Validation	**VI**	Software Verification and Validation	
VI.A	Theory	**VI.A**	Theory	Added Software as a Service (SaaS) and new product applications as examples.
VI.A.1	V&V methods	**VI.A.1**	V&V methods	
VI.A.2	Software product evaluation	**VI.A.2**	Software product evaluation	
VI.B	Test planning and design	**VI.B**	Test planning and design	Increased the cognitive level from Apply to Evaluate. Modified the subtext language to reflect the "Evaluate" cognitive level.
VI.B.1	Test strategies	**VI.B.1**	Test strategies	
VI.B.2	Test plans	**VI.B.2**	Test plans	
VI.B.3	Test designs	**VI.B.3**	Test designs	
VI.B.4	Software tests	**VI.B.4**	Software tests	
VI.B.5	Tests of external products	**VI.B.5**	Tests of external products	
VI.B.6	Test coverage specifications	**VI.B.6**	Test coverage specifications	
VI.B.7	Code coverage techniques	**VI.B.7**	Code coverage techniques	
VI.B.8	Test environments	**VI.B.8**	Test environments	Added manage.
VI.B.9	Test tools	**VI.B.9**	Test tools	
VI.B.10	Test data management	**VI.B.10**	Test data management	Added Software as a Service (SaaS) and new product applications as examples.
VI.C	Reviews and inspections	**VI.C**	Reviews and inspections	
VI.D	Test execution documents	**VI.D**	Test execution documents	

2016 BoK		2023 BoK		Notes
Code	**Label**	**Code**	**Label**	
VII	Software Configuration Management	**VII**	Software Configuration Management	Subtracted 4 questions
VII.A	Configuration infrastructure	**VII.A**	Configuration infrastructure	
VII.A.1	Configuration management team	**VII.A.1**	Configuration management team	
VII.A.2	Configuration management tools	**VII.A.2**	Configuration management tools	
VII.A.3	Library processes	**VII.A.3**	Library processes	
VII.B	Configuration identification	**VII.B**	Configuration identification	
VII.B.1	Configuration items	**VII.B.1**	Configuration items	
VII.B.2	Software builds and baselines	**VII.B.2**	Software builds and baselines	
VII.C	Configuration control and status accounting	**VII.C**	Configuration control and status accounting	
VII.C.1	Item change and version control	**VII.C.1**	Item change and version control	
VII.C.2	Configuration Control Board (CCB)	**VII.C.2**	Configuration Control Board (CCB)	
VII.C.3	Concurrent development	**VII.C.3**	Concurrent development	
VII.C.4	Status accounting	**VII.C.4**	Status accounting	Increased the cognitive level from Understand to Apply. Modified the subtext language to reflect the "Apply" cognitive level.
VII.D	Configuration audits	**VII.D**	Configuration audits	
VII.E	Product release and distribution	**VII.E**	Product release and distribution	
VII.E.1	Product release	**VII.E.1**	Product release	
VII.E.2	Customer deliverables	**VII.E.2**	Customer deliverables	
VII.E.3	Archival processes	**VII.E.3**	Archival processes	