DRAFT INTERNATIONAL STANDARD
ISO/IEC DIS 24773-4

ISO/IEC JTC **1**/SC **7**

Secretariat: **BIS**

Voting begins on:
2022-08-09

Voting terminates on:
2022-11-01

Software and Systems Engineering — Certification of software and systems engineering professionals —

Part 4:
Software engineering

ICS: 35.080

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC FDIS 24773-4
https://standards.iteh.ai/catalog/standards/sist/46a94daa-06ab-4be0-9cb8-
c3b5a94815be/iso-iec-fdis-24773-4

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 24773-4:2022(E)

© ISO/IEC 2022

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 24773 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information Technology, Subcommittee SC 7, Software and Systems Engineering.

Introduction

ISO/IEC 24773 (all parts) is a series which addresses the certification of professionals in software and systems engineering. ISO/IEC 24773-1 contains general requirements for such certification schemes. This document contains requirements specific to certification schemes for software engineering professionals.

The concepts, and requirements for certification schemes contained in ISO/IEC 24773-1 and ISO/IEC 17024 apply to this document.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC FDIS 24773-4
https://standards.iteh.ai/catalog/standards/sist/46a94daa-06ab-4be0-9cb8-
c3b5a94815be/iso-iec-fdis-24773-4

Software and Systems Engineering — Certification of software and systems engineering professionals —

Part 4:
Software engineering

1 Scope

This document elaborates requirements and recommendations for certifications schemes based on 24773-1, which are specific to the domain of software engineering.

2 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this standard.

ISO/IEC 17024:2012, *Conformity assessment — General requirements for bodies operating certification of persons*

ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*

ISO/IEC 24773-1:2019, *Software and systems engineering — Certification of software and systems engineering professionals — Part 1: General requirements*

3 Terms and Definitions

For the purposes of this document, unless superseded by definitions included within this section, the terms and definitions already defined in ISO/IEC 17024:2012, ISO/IEC TS 17027:2014, ISO/IEC 24773-1:2019, ISO/IEC 19759:2015 and ISO/IEC/IEEE 12207:2017 apply.

3.1
Knowledge Area
KA
An identified group of knowledge

3.2
software
computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system

[SOURCE: IEEE 828]

3.3
software engineering
systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software

[SOURCE: ISO/IEC/IEEE 24765]

4   Conformance

This document contains normative requirements for conformant certification schemes which are specific to Software Engineering. General requirements for conformant schemes also apply, and these are defined in ISO/IEC 24773-1:2019 and incorporated by reference into this document.

A certification scheme (the Scheme) for professionals in software engineering may claim conformance to this standard if it satisfies all requirements contained in this document; ISO/IEC 24773-1:2019; and ISO/IEC 17024:2012.

The following requirements, recommendations and associated criteria may also be used to compare certification and qualification schemes for professionals in software engineering.

5   Requirements for Certification of Software Engineering Professionals

5.1   General

This clause contains requirements and recommendations specific to the certification of Software Engineering professionals. The following requirements, recommendations and associated criteria may also be used to compare certification and qualification schemes of Software Engineering professionals.

5.2   Fundamental Components of a Conformant Scheme

This clause contains additional requirements for fundamental components defined in clause 6.3 of ISO/IEC 24773-1:2019.

5.2.1   The Scheme shall contain a description of software engineering professionals to be certified or targeted by that scheme.

5.2.2   A list of skills that the professional to be certified in software engineering is expected to exhibit shall be mapped by the Scheme to the knowledge areas of the reference body of knowledge (BOK), and to the defined competencies.

NOTE   A sample list of skills and competencies are shown in <u>Annex B</u> and <u>C</u> respectively. An exemplar mapping is shown in <u>Annex D</u>.

5.2.3   Each task or activity associated with the Scheme shall be "mapped" to at least one of the following:

— KA of the Guide to the SWEBOK, ISO/IEC TR 19759:2015[1];

— Process, activity or task of ISO/IEC/IEEE 12207:2017;

— Technical skill area of IEEE Software Engineering Competency Model (SWECOM) V1[2]

NOTE   The Scheme may define tasks at a different abstraction level of the tasks defined in ISO/IEC/IEEE 12207:2017.

NOTE   Tasks and their descriptions are associated with any certification scheme. Each scheme must declare a scope and describe the targeted individuals who would be certified in order to meet the requirements of ISO 17024. Part of the required scope and target description includes a list of tasks which represent the activities performed by certificants within the professional scope of that certification. These tasks are high level activity descriptions and not a comprehensive set of details of technical tasks which may be included within structure of the knowledge, skills or competencies of the scope.

5.2.4   The Scheme should require a minimum level of formal engineering education, where "formal engineering education" is defined as an engineering or computing degree from an accredited program,

and where "computing degree" is one of the following: computer science (CS), computer engineering (CE), software engineering (SE), information systems (IS) and information technology (IT).

NOTE   ISO/IEC 24773-1 contains a requirement that all conformant schemes define and document a minimum level of education, but this additional requirement is more specific regarding that minimum.

While minimum formal education requirements allow for various computing degrees, the requirements related to knowledge and coverage of Kas by the BOK (clause 5.3.3) used by the Scheme still apply.

5.2.5   If formal engineering education is required as part of the certification scheme, the requirements shall be clearly stated.

NOTE   The engineering degree does not have to be in software engineering, but must be followed by software engineering work/experience.

5.2.6   If on-the-job experience in software engineering is to be acceptable as part or total replacement for formal engineering education, the experience defined by the Scheme shall be objectively described and the verifiable criteria shall be made explicit. The verifiable on-the-job experience in software engineering should be at least as long in years as the formal engineering education, and twice as long (at least eight years) if no formal university engineering degree is required.

5.3   Knowledge

5.3.1   The reference body of knowledge is ISO/IEC 19759:2015 (SWEBOK)[1].

The reference body of knowledge defines the following knowledge areas:

— Software Requirements

— Software Design

— Software Construction

— Software Testing

— Software Maintenance

— Software Configuration Management

— Software Engineering Management

— Software Engineering Process

— Software Engineering Models and Methods

— Software Quality

— Software Engineering Professional Practice

— Software Engineering Economics

— Computing Foundations

— Mathematical Foundations

— Engineering Foundations

NOTE   An elaboration of software engineering knowledge areas is shown in Annex A.

3

5.3.2   The Scheme shall include requirements for knowledge of the following software engineering standards:

ISO/IEC/IEEE 12207:2017 Systems and Software Engineering – Software Life Cycle Processes

5.3.3   If the Scheme contains another BOK (i.e. "alternative BOK") other than the reference BOK identified in 5.3.1, the knowledge areas of the "alternative BOK" identified in the Scheme shall be mapped to knowledge areas of the reference BOK. The Scheme shall demonstrate that the "alternative BOK" covers the scope of the reference BOK as defined by the knowledge areas in 5.3.1, with rationale supporting any reduction in depth of coverage.

5.4   Skill

5.4.1   The Scheme shall define software engineering related skills covered by the Scheme, where they are specifically assessed or evaluated.

Skills covered by the scheme should be placed into groups or categories, where skills are related, and where there are a large number of skills in one group or category.

NOTE   An elaboration of software engineering skills is shown in Annex B.

5.4.2   The skills defined by the Scheme should be mapped to one or more competencies (5.5) defined and covered by the Scheme.

5.4.3   The Scheme should define performance levels for each skill so that the competencies defined by the Scheme can be achieved.

5.5   Competence

5.5.1   The Scheme shall identify software engineering specific competencies required for the software engineering professional to be certified. These may be drawn from the competencies defined in SWECOM 1.0[2] or sample competencies listed in Annex C.

NOTE   An elaboration of software engineering competencies is shown in Annex C.

5.5.2   The competencies associated with the Scheme shall define outcomes, deliverables or results from various software engineering tasks associated with that competency.

NOTE   Competencies may be grouped into competency areas or groups. The Scheme may define its own competency groups and competencies within them.

5.5.3   The reference life cycle process is ISO/IEC/IEEE 12207:2017.

5.5.4   The Scheme should define proficiency levels for each competency so that the tasks or activities defined by the Scheme can be achieved.

Annex A
(Informative)

Elaboration of Software Engineering Knowledge Areas

Table A.1 lists and provides some elaboration regarding the Knowledge Areas contained in the Reference Body of Knowledge as defined in Clause 5.3.

The source document (ISO/IEC TR 19759:2015) contains detailed information regarding sub-topics and supporting literature for each Knowledge Area.

Table A.1 — Elaboration of Software Engineering Knowledge Areas

Software Engineering Knowledge Areas	Sub-Areas and Sub-Topics
Software Requirements	Software Requirements Fundamentals, Requirements Process, Requirements Elicitation, Requirements Analysis, Requirements Specification, Requirements Validation, Requirements Management, Requirements tools.
Software Design	General Design Concepts, Software Design Process, Software Design Principles, Key Issues in Software Design (Concurrency, Control and Handling of Events, Data Persistence, Distribution of Components, Error and Exception Handling and Fault Tolerance, User Interaction and Presentation, Security), Software Structure and Architecture, User Interface Design, Software Design Quality Analysis and Evaluation, Quality Attributes, Software Design Description and Notations, Software Design Strategies and Methods, Software Design tools.
Software Construction	Construction principles and objectives, reuse, complexity, construction standards, managing construction (planning and measurement), Integration, construction technologies, software construction tools.
Software Testing	Testing-Related Concepts and Terminology, Test Techniques, Test-Related Measures, Test Process, Software Testing tools.
Software Maintenance	Software Maintenance Fundamentals Definitions and Terminology, software evolution, maintenance costs, maintenance process, techniques for maintenance, software maintenance tools.
Software Configuration Management	Management of the SCM Process, Software Configuration Identification, Software Configuration Control, Software Configuration Status Accounting, Software Configuration Auditing, Software Release Management and Delivery, Software Building, Software Configuration Management tools.
Software Engineering Management	Project Initiation and Scope Definition, Software Project Planning, Risk Management, Software Project Enactment, Software Acquisition and Supplier Contract Management, Implementation of Measurement Process, Project Review and Evaluation, Project Closure, Software Engineering Measurement, Software Engineering Management tools.
Software Engineering Process	Software Lifecycles, Software Process Definition, Software Process Assessment and Improvement, Software Process and Product Measurement, Software Process Measurement Techniques, Software Engineering Process tools.
Software Engineering Methods	Models and Modeling Principles, Analysis of Models, Software Engineering Methods, formal methods, agile methods.
Software Quality	Software Quality Fundamentals, Costs of Quality, Models and Quality Characteristics, Software Safety, Software Quality Management Processes, Software Quality Assurance, Verification & Validation, Reviews and Audits, Software Quality Requirements, Defect Characterization, Software Quality Management Techniques, Software Quality Measurement, Software Quality tools.