

Evaluating the Software Engineering Curriculum at JUST: A Comparative Analysis with IEEE Knowledge Areas

Moh'd A. Radaideh¹ and Qasem Abu Al-Haija ^{*2}

Submitted:12/03/2024 Revised: 27/04/2024 Accepted: 04/05/2024

Abstract: Contribution: This paper examines the compliance of the Software Engineering (SE) program at Jordan University of Science and Technology (JUST) with the fifteen Software Engineering Knowledge Areas (SEKAs) established in the IEEE-CS SWEBOK. This research is the first to determine the scope of these knowledge fields in a SE undergraduate program curriculum.

Background: Although the Institute of Engineering and Technology (IET) offers the SE undergraduate program at JUST, aligning it with the IEEE view of SE is essential.

Research Questions: This research aims to answer some questions, such as how much the SEKAs' topics are included in JUST's SE undergraduate program. How to eliminate the identified coverage gap.

Methodology: Observe the coverage of the SEKAs' topics in the SE undergraduate program curriculum courses at JUST. Then, accordingly CLASSIFY theSEKAs into (a) Full-Compliance (e.g., when concerned Knowledge Area is fully covered across one or more of the SE undergraduate program curriculum' courses); (b) High-Compliance (e.g., when concerned Knowledge Area is highly covered); (c) Partial-Compliance (e.g., when concerned Knowledge Area is partially covered); or (d) Poor-Compliance (e.g., when concerned Knowledge Area is poorly covered).

Findings: This research concluded that the SE undergraduate program complies with the Software Requirements (SW-RQTs), Software Testing (SW-TS), SE Management (SW-MG), and Software Quality Knowledge Areas (SQKAs). The Software Design (SW-DS), SE Models and Methods, Computing Foundations (CFs), and Mathematical Foundations Knowledge Areas (MFKAs). c) Ensure that the software construction (SW-CN), maintenance (SW-MN), configuration management (SW-CM), and SE process knowledge areas are consistent and compliant. d) Unsatisfactory in the SE Professional Practice, SE Economics, and Engineering Foundation Knowledge Areas (EFKAs).

Keywords: Unsatisfactory, Professional, Computing Foundations, Software Requirements, Knowledge

1. Introduction

The Undergraduate Program in SE (SEUP) offered by JUST is accredited by the Institute of Engineering and Technology (IET). However, further expansion is required to ensure its readiness for additional accreditation, such as ABET. The discipline of SE is divided into fifteen Knowledge Areas, also known as SEKA, according to IEEE-CS SWEBOK. The SEUP's specialization courses are listed in Table 1.

This paper discusses the need for more coverage of the fifteen SEKAs in the SE undergraduate curriculum (SEUP) courses at JUST. These fifteen SEKAs are:

- 1- SEKA#1 (SW-RQTs). This Knowledge Area (e.g., chapter 1 of the SWEBOK) concerns the SW-RQTs' Elicitation, Analysis, Specification, Validation, and Management throughout the Software Product's Lifecycle [13, 19-26].
- 2- SEKA#2 (SW-DS). This Knowledge Area (e.g., chapter 2 of the SWEBOK) is about specifying the targeted Software Product per its Requirements [13, 27-29].
- 3- SEKA#3 (SW-CN). This Knowledge Area (e.g., chapter 3 of the SWEBOK) is concerned with developing and constructing the targeted Software Product by its Design and Requirements [13, 30-34].
- 4- SEKA#4 (SW-TS). This Knowledge Area (e.g., chapter 4 of the SWEBOK) is concerned with assessing the Outputs of the Software Product in comparison with its expected outputs [13, 35-40].
- 5- SEKA#5 (SW-MN). This Knowledge Area (e.g., chapter 5 of the SWEBOK) handles any new

Table 1. The SE Courses At JUST

SE210 Java Programming
SE220 Software Modelling
SE230 Fundamentals of Software Engineering II
SE310 Visual Programming
SE320 Systems Analysis and Design
SE321 Software Requirements Engineering
SE323 Software Documentation
SE324 Software Architecture and Design
SE430 Software Testing
SE431 Software Security
SE432 Software Engineering for Web Applications
SE440 Project Management
CS318 Human-Computer Interaction
SE471 Client/Server Programming
SE326 Software Engineering Laboratory
SE441 Software Quality Assurance (Elective)

¹ Department of Software Engineering, Jordan University of Science and Technology, PO Box 3030, Irbid 22110, Jordan
ORCID ID : 0000-0002-3900-9710

² Department of Cybersecurity, Jordan University of Science and Technology, PO Box 3030, Irbid 22110, Jordan
ORCID ID : 0000-0003-2422-0297

* Corresponding Author Email: qabuhaija@just.edu.jo

Requirements or Changes in the Software Product's Operational Environment [13, 41].

- 6- SEKA#6 (SW-CM). This Knowledge Area (*e.g., chapter 6 of the SWEBOK*) concerns Configurational needs throughout the Software Product's Lifecycle [13, 42].
- 7- SEKA#7 (SE Management). This Knowledge Area (*e.g., chapter 7 of the SWEBOK*) concerns the IEEE-CS Software Project Management Approach [13, 43-50].
- 8- SEKA#8 (SE Process). This Knowledge Area (*e.g., chapter 8 of the SWEBOK*) concerns the various Groups of Processes for SE [13, 51-67].
- 9- SEKA#9 (SE Models and Methods). This Knowledge Area (*e.g., chapter 9 of the SWEBOK*) concerns the various Groups of Models and Methods for SE [13, 68-80].
- 10- SEKA#10 (Software Quality). This Knowledge Area (*e.g., chapter 10 of the SWEBOK*) concerns the various Aspects of Software Quality Assurance and Control Management [13, 81-98].
- 11- SEKA#11 (SE Professional Practice). This Knowledge Area (*e.g., chapter 11 of the SWEBOK*) concerns the SE Professional Aspects of Professionalism, Team Work, and Communication Skills [13, 99-100].
- 12- SEKA#12: SE Economics. This Knowledge Area (*e.g., chapter 12 of the SWEBOK*) concerns the various Aspects of SE Economics [13].
- 13- SEKA#13 (CFs). This Knowledge Area (*e.g., chapter 13 of the SWEBOK*) concerns the various Computing Topics required for Software Engineering, including Programming, Databases, Networking, Computer Architecture, and so forth [13].
- 14- SEKA#14 (Mathematical Foundations). This Knowledge Area (*e.g., chapter 14 of the SWEBOK*) concerns the various Mathematical Topics required for SE [13].
- 15- SEKA#15 (Engineering Foundations). This Knowledge Area (*e.g., chapter 15 of the SWEBOK*) concerns the various Engineering Aspects required for SE [13].

This research work found that the SE undergraduate program is in:

Please find the revised text below:

- 1- Complete adherence to the SW-RQTs, SW-TS, SW-MG (Software Project Management), and SQKAs.
- 2- High-level adherence to the SW-DS, SE Models and Methods, CFs, and MFKAs.
- 3- Partial adherence to the SW-CN, SW-MN, SW-CM, and SE Process Knowledge Areas.

4- Low level of adherence to the SE Professional Practice, SE Economics, and EFKAs.

The paper is structured into several sections. The Introduction is presented in Section 1, while Section 2 elaborates on the related work. Section 3 details our research approach, and Section 4 covers the SEKAs included in the undergraduate program at JUST. The research findings and recommendations are summarized in Section 5, and Section 6 concludes the paper.

2. Related Work

This section elaborates on many relevant SE Education publications. Garousi et al. [10-11] examined the collaboration between the Software industry and SE Academic Education to address the gap between the two communities. The authors concluded that collaboration between the two communities was inadequate, b) identified the various challenges causing such collaboration, and (c) recommended some strategies to overcome these challenges. Additionally, they analyzed the challenges new SE graduates face in their work. The authors claimed that such challenges were due to misalignment of the skills gained during their undergraduate studies.

Meziane et al. [12] compared the Computer Science and SE Programs from their Knowledge Areas and concluded that numerous variations exist.

Fox et al. [9] participated in teaching SE as Massive Open Online Courses (MOOCs) and Small Private Online Courses (SPOCs) to develop a SE curriculum. The authors asserted that MOOCs and SPOCs can improve the quality of SE courses, given that challenges can be discussed with numerous involved participants. The authors also stated six SE Education Challenges. (i) Students need more time to study the material they are given. (ii) SE Faculty members typically need more industrial expertise. (iii) As there are many SE methods, it takes time to choose and focus on one of them. (iv) There are few practical textbooks on SE out there. (v) Tools that support SE Methods are typically expensive. Thus, Academic institutions need help to host and deploy them, and the Industry always complains about the quality of SE Education.

In their study, Qiu et al. [18] investigated the effectiveness of a problem-based learning approach using a blend of face-to-face and e-learning environments in teaching SE principles and skills to undergraduate students. The authors reported that most participating students found this approach comfortable and performed better than expected.

Bastarrica et al. [6] surveyed the importance and challenges in the SE students' projects. The authors found that the participating students' soft skills increased and accumulated while their technical challenges decreased. In addition, they indicated that the participating students stated that planning

their projects and collaboration were more problematic than they anticipated.

Barzilay et al. [5] proposed a comprehensive SE course that included SE Fundamentals, Practices and Tools, Productization, and Technology Evolution.

Dekhane et al. [8] emphasized the integration of projects of different domains into one interdisciplinary project. The authors analyzed their proposed work by providing SE students with authentic experiences.

Daimi et al. [7] demonstrated a model for SE courses. Their model encompasses brainstorming, critical thinking, case methods, problem-based learning, trimming techniques, and opportunity recognition.

This research is different from the related works mentioned above as it assesses the compliance of the SE curriculum at JUST with the SEKAs. It measures how much the curriculum covers these topics and expected learning outcomes. This research is unique as it is the first attempt to measure the coverage of SEKAs in any SE curriculum in terms of its contents and learning outcomes. References [14-17] support this claim.

3. Research Methodology

The methodology employed in conducting this research study is outlined as follows:

- 1- Separating the SEKAs into two groups.
 - SE Specialization-Realized Knowledge Areas (e.g., SEKAs#1-10).
 - SE Professional-Support-Relatable Knowledge Areas (e.g. SEKAs#10-15).
- 2- Examining the coverage of the SEKAs in the SE Program Curriculum of JUST.
 - Observing the coverage of the Specialization-Related Knowledge Areas topics in the Specialization courses in the curriculum.
 - Inquiring about topics related to professional-support-related knowledge areas in the university, along with the college's required courses in the curriculum.
- 3- Classifying the coverage of each SEKA in the curriculum into one of the following levels:
 - Full-Compliance. This indicates a full coverage of the concerned SEKA across one or more of the courses of the SE Curriculum.
 - High-Compliance. This indicates a high coverage of the concerned SEKA across one or more of the courses of the SE curriculum.

- Partial-Compliance. This indicates a partial coverage of the concerned SEKA across one or more SE curriculum courses.
 - Poor-Compliance. This indicates poor coverage of the concerned SEKA across one or more SE curriculum courses.
- 4- Analyzing the Integration of SEKA Main Topics within Course Learning Outcomes (CLOs): This stage involves categorizing the extent to which the core topics of each SEKA (SEKA) are covered in the curriculum's course learning outcomes (CLOs). The coverage of CLOs about each SEKA is evaluated and classified into distinct levels:
 - Full-Compliance. This indicates a complete declaration of all of the main topics of the SEKA as course learning outcomes (CLOs) across the courses of the SE Curriculum.
 - High-Compliance. This indicates a declaration of most of the major topics of the SEKA as course learning outcomes (CLOs) across the courses of the SE Curriculum.
 - Partial-Compliance. This indicates a declaration of some of the primary topics of the SEKA as course learning outcomes (CLOs) across the courses of the SE Curriculum.
 - Poor-Compliance. This indicates a declaration of none or a few of the main topics of the SEKA as course learning outcomes (CLOs) in the courses of the SE curriculum.
 - 5- Identifying the coverage gaps and making recommendations.
 - 6- Validating the research's primary objective's success to facilitate the potential ABET Certification of the SE Undergraduate Program of JUST.

4. Results (SEKAs Coverage in the SE-Curriculum at JUST)

4.1 Coverage of the SEKA#1 (SW-RQTs)

SEKA#1 (SW-RQTs) appears fully covered in the SE Curriculum at JUST in the SE230 Fundamental of Software Engineering, SE321 SE Requirements, and SE430 SW-TS courses. Additionally, the main topics of this Knowledge Area are referred to as learning outcomes (CLOs) in the same courses.

Table 2. Coverage of SEKA#1 in the SE Curriculum at JUST

Topics of the Software Requirements Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Requirements Fundamentals (Definition of a Software Requirement, Product and Process Requirements, Functional and Non-functional Requirements, Emergent Properties, System Requirements and Software Requirements, and Quantifiable Requirements)	- SE230 - SE321	- SE230 (CLO2) - SE321 (CLO1/ CLO4)
Topic#2: Requirements Process (Process Models, Process Actors, Process Support and Management, and Process Quality and Improvement)	- SE230 - SE321	- SE321 (CLO1)
Topic#3: Requirements Elicitation (Requirements Sources and Elicitation Techniques)	- SE321	- SE321 (CLO3)
Topic#4: Requirements Analysis (Requirements Classification, Conceptual Modelling, Architectural Design and Requirements Allocation, Requirements Negotiation, and Formal Analysis)	- SE321	- SE321 (CLO2)
Topic#5: Requirements Specification (System Definition Document, System Requirements Specification, and Software Requirements Specification)	- SE321 - SE440	- SE321 (CLO4)
Topic#6: Requirements Validation (Requirements Reviews, Prototyping, Model Validation, and Acceptance Tests)	- SE321 - SE440 - SE441	- SE321 (CLO7/ CLO8/ CLO9)
Topic#7: Practical Considerations (Iterative Nature of the Requirements Process, Change Management, Requirements Attributes, Requirements Tracing, and Measuring Requirements)	- SE321 - SE430	- SE321 (CLO6)
Topic#8: Software Requirements Tools	- SE321	- SE321 (CLO9)

4.1 Coverage of the SEKA#2 (SW-DS)

Table III illustrates that SEKA#2 (SW-DS) appears to be comprehensively addressed within the SE-Curriculum at JUST across several courses, including SE210 Java Programming, SE220 Software Modelling, SE230 Fundamentals of Software Engineering, SE310 Visual Programming, SE324 Software Architecture and Design, SE432 SE for Web Applications, and CS318 Human-Computer Interaction. Moreover, the primary topics of this Knowledge Area are explicitly identified as learning outcomes (CLOs) within these courses.

Table 3. Coverage of SEKA#2 In the SE Curriculum at JUST

Topics of the Software Design Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Design Fundamentals (General Design Concepts, Context of Software Design, Software Design Process, and Software Design Principles)	- SE230 - SE324	- SE230 (CLO4) - SE324 (CLO1)
Topic#2: Key Issues in Software Design (Concurrency, Control and Handling of Events, Data Persistence, Distribution of Components, Error and Exception Handling and Fault Tolerance, Interaction and Presentation, and Security)	- SE210 - SE310 - SE324 - SE432	- SE210 (CLO6) - SE310 (CLO5) - SE324 (CLO1)
Topic#3: Software Structure and Architecture (Architectural Structures, Architectural Styles, Design Patterns, Design Decisions, etc.)	- SE324	- SE324 (CLO2)
Topic#4: User Interface Design (General User Interface Design Principles, User Interface Design Issues, The Design of User Interaction Modalities, The Design of Information Presentation, User Interface Design Process, Localization and Internationalization, and Metaphors and Conceptual Models)	- CS318	- CS318 (CLO6)
Topic#5: Software Design Quality Analysis and Evaluation (Quality Attributes, Quality Analysis and Evaluation Techniques, and Measures)	- SE324	- SE324 (CLO2)
Topic#6: Software Design Notations (Structural Descriptions - Static View, and Behavioural Descriptions - Dynamic View)	- SE324	- SE324 (CLO3)
Topic#7: Software Design Strategies and Methods (General Strategies, Function-Oriented Structured Design, Object-Oriented Design, Data Structure-Centered Design, and Component-Based Design)	- SE324	- SE324 (CLO3)
Topic#8: Software Design Tools	- SE324	- SE324 (CLO3)

4.1 Coverage of the SEKA#3 (SW-CN)

Table IV suggests that SEKA#3 (SW-CN) is partially addressed within the SE-Curriculum at JUST, notably in courses such as SE210 Java Programming, SE326 SE Lab 1, SE230 Fundamentals of SE, SE310 C# Visual Programming, SE321 SE Requirements, SE324 Software Architecture and Design, SE430 SW-TS, SE440 Software Project Management, SE441 Software Quality Assurance, and SE371 Client-Server. Furthermore, several key topics from this Knowledge Area are explicitly stated as learning outcomes (CLOs) in specific courses, including SE210 Java Programming, SE310 Visual Programming, SE430 SW-TS, and SE440 Software Project Management.

Table 4. The Coverage of SEKA#3 in the SE Curriculum at JUST

Topics of the Software Construction Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Construction Fundamentals (Minimizing Complexity, Anticipating Change, Constructing for Verification, Reuse, and Standards in Construction)	- SE440 - SE441	- SE440 (CLO4)
Topic#2: Managing Construction (Construction in Life Cycle Models, Construction Planning, and Construction Measurement)		
Topic#3: Practical Considerations (Construction Design, Construction Languages, Coding, Construction Testing, Construction for Reuse, Construction with Reuse, Construction Quality, and Integration)	- SE310 - SE310 - SE430 - SE441	- SE210 (CLO1) - SE310 (CLO1) - SE430 (CLO1/ CLO2)
Topic#4: Construction Technologies (API Design and Use; Object-Oriented Runtime Issues; Parameterization and Generics; Assertions, Design by Contract, and Defensive Programming; Error Handling, Exception Handling, and Fault Tolerance for Distributed Software; Executable Models; State-Based and Table-Driven Construction Techniques; Runtime Configuration and Internationalization; Grammar-Based Input Processing; Concurrency Primitives; Middleware; Construction Methods; Constructing Heterogeneous Systems; Performance Analysis and Tuning; Platform Standards; and Test-First Programming)	- SE210 - SE310 - SE324 - SE326 - SE371 - SE430	- SE210 (CLO3) - SE310 (CLO4/ CLO5)
Topic#5: Software Construction Tools (Development Environments; GUI Builders; Unit Testing Tools; and Profiling, Performance Analysis, and Slicing Tools)	- SE310 - SE326 - SE430	- SE310 (CLO3)

4.2 Coverage of the SEKA#4 (SW-TS)

SEKA#4 (SW-TS) appears to be fully covered in the SE Curriculum at JUST in the SE230 Fundamental of Software Engineering, SE430 SW-TS, and SE441 Software Quality Assurance courses. Additionally, the main topics of this Knowledge Area are referred to as learning outcomes (CLOs) in the same courses.

Table 5. The Coverage of SEKA#4 in The SE Curriculum at JUST

Topics of the Software Testing Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Testing Fundamentals (Testing-Related Terminology, Definitions of Testing and Related Terminology, Faults vs. Failures, and Relationship of Testing to Other Activities)	- SE430 - SE230 - SE441	- SE230 (CLO5) - SE430 (CLO1)
Topic#2: Test Levels (Topic#2.1: Test Targets (Unit Testing, Integration Testing, and System Testing); (Topic#2.2: Test Objectives (Acceptance / Qualification Testing, Installation Testing, Alpha and Beta Testing, Reliability Achievement and Evaluation, Regression Testing, Performance Testing, Security Testing, Stress Testing, Back-to-Back Testing, Recovery Testing, Interface Testing, Configuration Testing, and Usability and Human-Computer Interaction Testing)	- SE230 - SE430 - SE441 - CS318	- SE430 (CLO1/ CLO2) - SE441 - CS318 (CLOs)
Topic#3: Test Techniques (Based on the Software Engineer's Intuition and Experience, Input Domain-Based Techniques, Code-Based Techniques, Fault-Based Techniques, Usage-Based Techniques, Model-Based Testing Techniques, Techniques Based on the Nature of the Application, and Selecting and Combining Techniques)	- SE430 - SE441	- SE430 (CLO2/ CLO3)
Topic#4: Test-Related Measures (Evaluation of the Program Under Test and Evaluation of the Tests Performed)	- SE430	- SE430 (CLO4)
Topic#5: Test Process (Practical Considerations and Test Activities)	- SE430	- SE430 (CLO6)
Topic#6: Software Testing Tools (Testing Tool Support and Categories of Tools)	- SE430 - SE441	- SE430 (CLO5)

4.3 Coverage of the SEKA#5 (SW-MN)

SEKA#5 (SW-MN) appears to be partially covered in the SE Curriculum at JUST in the SE230 Fundamental of SE and SE441 Software Quality Assurance courses. Additionally, none of the main topics of this Knowledge Area are deemed to be learning outcomes (CLOs) in these courses.

Table VI. The Coverage of SEKA#5 in the SE Curriculum at JUST

Topics of the Software Maintenance Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Maintenance Fundamentals (Covered sub-topics are Definitions and Terminology, Nature of Maintenance, Need for Maintenance, and Majority of Maintenance Costs; Not-Covered sub-topics are Evolution of Software and Categories of Maintenance)	- SE230 - SE441	
Topic#2: Key Issues in Software Maintenance (Covered sub-topics are none; Not-covered sub-topics are Technical Issues, Management Issues, Maintenance Cost Estimation, Software Maintenance Measurement, Maintenance Processes, and Maintenance Activities)		
Topic#4: Techniques for Maintenance (Covered sub-topics are none; Not-covered sub-topics are Program Comprehension, Re-Engineering, Reverse Engineering, Migration, and Retirement)		
Topic#5: Software Maintenance Tools		

4.4 Coverage of the SEKA#6 (SW-CM)

TABLE VII indicates that SEKA#6 (SW-CM) is partially covered in the SE441 Software Quality Assurance course that covers the various SW-CM topics, but without going into details. Additionally, none of the main topics of this Knowledge Area are deemed to be learning outcomes (CLOs) in this course.

Table 6. The Coverage of SEKA#6 in the SE Curriculum at JUST

Topics of the Software Configuration Management Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Management of the SCM Process (Organizational Context for SCM, and Constraints and Guidance for the SCM Process)	- SE441	-
Topic#2: Planning for SCM (SCM Plan, Software Configuration Management Surveillance, Software Config. Identification, Identifying Items to Be Controlled, and Software Library)	- SE441	-
Topic#3: Software Configuration (Requesting, Evaluating, and Approving Software Changes, Implementing Software Changes, and Deviations and Waivers)	- SE441	-
Topic#4: Software Configuration Status Accounting (Software Configuration Status Information and Software Configuration Status Reporting)	- SE441	-
Topic#5: Software Configuration (Software Functional Configuration Audit, Software Physical Configuration Audit, and In-Process Audits of a Software Baseline)	- SE441	-
Topic#6: Software Release Management and Delivery (Software Building and Software Release Management Auditing)	- SE441	-
Topic#7: Software Configuration Management Tools	- SE441	-

4.5 Coverage of the SEKA#7 (SE Management)

TABLE VIII indicates that SEKA#7 (SE Management) appears to be fully supervised in the SE440 course. Furthermore, the main topics of this Knowledge Area are referred to as learning outcomes (CLOs) in this course.

Table 6. The Coverage of SEKA#7 in the SE Curriculum at JUST

Topics of the Software Engineering Management Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Initiation and Scope Definition (Determination and Negotiation of Requirements, Feasibility Analysis, and Process for the Review and Revision of Requirements)	- SE 440	- SE440 (CLOs)
Topic#2: Software Project Planning (Process Planning; Determine Deliverables, Effort, Schedule, and Cost Estimation; Resource Allocation; Risk Management; Quality Management; and Plan Management)	- SE 440	
Topic#3: Software Project Enactment (Implementation of Plans, Software Acquisition and Supplier Contract Management, Implementation of Measurement Process, Monitor Process, Control Process, and Reporting)	- SE 440	
Topic#4: Review and Evaluation (Determining Satisfaction of Requirements and Reviewing and Evaluating Performance)	- SE 440	
Topic#5: Closure (Determining Closure and Closure Activities)	- SE 440	
Topic#6: Software Engineering Measurement (Establish and Sustain Measurement Commitment, Plan the Measurement Process, Perform the Measurement Process, Evaluate Measurement, Software Engineering Management Tools, and Matrix of Topics vs. Reference Material)	- SE 440	

4.6 Coverage of the SEKA#8 (SE Process)

Table IX reveals that SEKA#8 (SE Process) is partially addressed within the SE-Curriculum at JUST, particularly in courses such as SE230 Fundamentals of Software Engineering, SE324 Software Architecture and Design, SE323 Software Documentation, SE440 Software Project Management, and SE441 Software Quality Assurance. Moreover, certain key topics from this Knowledge Area are explicitly outlined as learning outcomes (CLOs) in these respective courses.

Table 8. Coverage of SEKA#8 in the SE Curriculum at JUST

Topics of the Software Process Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Process Definition (Software Process Management and Software Process Infrastructure)	- SE230 - SE324 - SE441	-
Topic#2: Software Life Cycles (Categories of Software Processes, Software Life Cycle Models, Software Process Adaptation, and Practical Considerations)	- SE230 - SE324 - SE430 - SE440	- SE440 (CLO4)
Topic#3: Software Process Assessment and Improvement (Software Process Assessment Models, Software Process Assessment Methods, Software Process Improvement Models, and Continuous and Staged Software Process Ratings)	- SE230 - SE440	- SE440 (CLO4)
Topic#4: Software Measurement (Software Process and Product Measurement, Quality of Measurement Results, Software Information Models, and Software Process Measurement Techniques)	- SE230 - SE323 - SE324 - SE441	-
Topic#5: Software Engineering Process Tools	- SE440	- SE440 (CLO4)

4.7 Coverage of the SEKA#9 (SE Models and Methods)

Table X demonstrates that SEKA#9 (SE Models and Methods) is partially addressed within the SE-Curriculum at JUST, notably in courses such as SE220 Software Modelling, SE321 SE Requirements, SE324 Software Architecture and Design, SE440 Software Project Management, and SE441 Software Quality Assurance.

Additionally, the primary topics of this Knowledge Area are integrated into the learning outcomes (CLOs) of the SE220 and SE440 courses, further reinforcing their coverage within the curriculum.

TABLE 9: Coverage of the SEKA#9 in the SE Curriculum at JUST

Topics of the Software Engineering Models and Methods Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Modelling (Modelling Principles; Properties and Expression of Models; Syntax, Semantics, and Pragmatics; and Pre-conditions, Post-conditions, and Invariants)	- SE220	- SE220 (CLOs)
Topic#2: Types of Models (Information Modelling, Behavioural Modelling, and Structure Modelling)	- SE220	-
Topic#3: Analysis of Models (Analysing for Completeness, Analysis for Consistency, Analysis for Correctness, Traceability, and Interaction Analysis)	- SE220/ SE321 - SE324/ SE440 - SE441	-
Topic#4: Software Engineering Methods (Heuristic Methods, Formal Methods, Prototyping Methods, and Agile Methods)	- SE321 - SE440	- SE440 (CLO2)

4.8 Coverage of the SEKA#10 (Software Quality)

Table XI states that SEKA#10 (Software Quality) appears to be fully covered in the SE430 SW-TS and SE441 Software Quality Assurance courses. Additionally, the main topics of this Knowledge Area are referred to as learning outcomes (CLOs) in the SE441 course.

Table 10. COVERAGE OF SEKA#10 IN SE CURRICULUM AT JUST

Topics of the Software Quality Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Quality Fundamentals (Software Engineering Culture and Ethics, Value and Costs of Quality, Models and Quality Characteristics, Software Quality Improvement, and Software Safety)	- SE441	- SE441 (CLOs)
Topic#2: Software Quality Management Processes (Software Quality Assurance, Verification & Validation, and Reviews and Audits)	- SE441	
Topic#3: Practical Considerations (Software Quality Requirements, Defect Characterization, Software Quality Management Techniques, and Software Quality Measurement)	- SE441	
Topic#4: Software Quality Tools	- SE441	

4.9 Coverage of the SEKA#11 (SE Professional Practice)

Table XII indicates that SEKA#11 (SE Professional Practice) needs to be improved in the SE440 course. Additionally, none of the main topics of this Knowledge Area are declared as learning outcomes (CLO) in this course.

Table XII: Coverage of SEKA#11 in the SE Curriculum at JUST

Topics of the Software Engineering Professional Practice Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Professionalism (Accreditation, Certification, and Licensing; Codes of Ethics and Professional Conduct; Nature and Role of Professional Societies; Nature and Role of Software Engineering Standards; Economic Impact of Software; Employment Contracts; Legal Issues; Documentation; and Trade-off Analysis)	- SE440	- SE440 (-20% of the topics are covered in this course)
Topic#2: Group Dynamics and Psychology (Dynamics of Working in Teams, Individual Cognition, Dealing with Problem Complexity, Interacting with Stakeholders, Dealing with Uncertainty and Ambiguity, and Dealing with Multicultural Environments)	-	
Topic#3: Communication Skills (Reading, Understanding, and Summarizing Writing; Team and Group Communication; and Presentation Skills)	-	

4.10 Coverage of the SEKA#12 (SE Economics)

TABLE XIII indicates that SEKA#12 (SE Economics) needs to improve in the SE440 course. Additionally, none of the main topics of this Knowledge Area are deemed to be learning outcomes (CLO) in this course.

Table XIII: Coverage of the SEKA#12 in the SE Curriculum at JUST

Topics of the Software Engineering Economics Knowledge Area	Covering Courses	Covering CLOs
Topic#1: Software Engineering Economics Fundamentals (Finance, Accounting, Controlling, Cash Flow, Decision-Making Process, Valuation, Inflation, Depreciation, Taxation, Time-Value of Money, Efficiency, Effectiveness, and Productivity)	SE440 (~20% of the topics are covered in this course)	
Topic#2: Life Cycle Economics (Product, Project, Program, Portfolio, Product Life Cycle, Project Life Cycle, Proposals, Investment Decisions, Planning Horizon, Price and Pricing, Cost and Costing, Performance Measurement, Earned Value Management, Termination Decisions, and Replacement and Retirement Decisions)		
Topic#3: Risk and Uncertainty (Goals, Estimates, and Plans; Estimation Techniques; Addressing Uncertainty; Prioritization; Decisions under Risk; and Decisions under Uncertainty)		
Topic#4: Economic Analysis Methods (For-profit Return on Investment Decision Analysis, Minimum Acceptable Rate of Return, Return on Capital Employed, Cost-Benefit Analysis, Cost-Effectiveness Analysis, Break-Even Analysis; Business Case, Multiple Attribute Evaluation, and Optimization Analysis)		
Topic#5: Practical Considerations (The "Good Enough" Principle; Friction-Free Economy; Ecosystems; and Offshoring and Outsourcing)		

4.11 Coverage of the SEKA#13 (CFs)

Table XIV highlights that SEKA#13 (CFs) is extensively covered within the SE-Curriculum at JUST across various courses, including SE210 Java Programming, SE225 SE Lab, SE230 Fundamentals of Software Engineering, SE310 C# Visual Programming, SE321 SE Requirements, SE322 Software Design, SE430 SW-TS, SE440 Software Project Management, SE441 Software Quality Assurance, and SE471 Client-Server.

Table XIV: Coverage of SEKA#13 in the SE Curriculum at JUST

Topics of the Computing Foundations Knowledge Area	Covering Courses	Covering CLOs
Subject#1: Problem-Solving Techniques	SE210 / SE322 / SE310	All of these subjects contribute to the various learning outcomes of the Software Engineering undergraduate program at JUST.
Subject#2: Abstraction		
Subject#3: Programming Fundamentals	SE210 / SE310	
Subject#4: Programming Language Basics	SE210 / SE310	
Subject#5: Debugging Tools and Techniques	SE210 / SE310 / SE430 / SE441	
Subject#6: Data Structure and Representation	SE310 / SE210	
Subject#7: Algorithms and Complexity		
Subject#8: Basic Concept of a System	SE320	
Subject#9: Computer Organization	SE320 / SE430 / SE322	
Subject#10: Compiler Basics		
Subject#11: Operating Systems Basics		
Subject#12: Database Basics and Data Management	SE310 / SE320 / SE471	
Subject#13: Network Communication Basics	SE471	
Subject#14: Parallel and Distributed Computing	SE322	
Subject#15: Basic User Human Factors		
Subject#16: Basic Developer Human Factors		
Subject#17: Secure Software Development and Maintenance	SE431	

4.12 Coverage of the SEKA#14 (Mathematical Foundations)

SEKA#14 (Mathematical Foundations) appears highly regarded in the MATH241 Discrete Mathematics course.

Table XV: Coverage of the SEKA#14 the SE-Curriculum at JUST

Topics of the Mathematical Foundations Knowledge Area	Covering Courses	Covering CLOs
Subject#1: Sets Operations; Relations; and Relations and Functions	MATH241 covers the majority of these topics.	
Subject#2: Propositional Logic; and Predicate Logic		
Subject#3: Proof Techniques (Methods of Proving Theorems)		
Subject#3: Basics of Counting		
Subject#4: Graphs and Trees		
Subject#5: Discrete Probability		
Subject#6: Finite State Machines		
Subject#7: Grammars and Language Recognition.		
Subject#8: Numerical Precision, Accuracy, and Errors		
Subject#9: Number Theory (Divisibility and Prime Numbers)		
Subject#10 Algebraic Structures (Groups and Rings)		

4.13 Coverage of the SEKA#15 (EFKAs)

TABLE XVI indicates that the JUST curriculum does not include SEKA#15 (EFKAs).

Table XVI: Coverage of SEKA#15 in the SE Curriculum at JUST

Topics of the Mathematical Foundations Knowledge Area	Covering Courses	Covering CLOs
Subject#1: Empirical Methods and Experimental Techniques	The Software Engineering undergraduate curriculum does not seem to cover these topics.	
Subject#2: Statistical Analysis		
Subject#3: Measurement		
Subject#4: Engineering Design		
Subject#5: Modelling, Simulation, and Prototyping		
Subject#6: Standards		
Subject#7: Root Cause Analysis		

5. Discussion and Recommendations

This research study examined the compliance of the SE curriculum with the SEKAs of the IEEE-CS SWEBOOK. TABLE XVII provides a comprehensive overview of these in the SE Curriculum at JUST.

As illustrated in TABLE XVII, the degree of compliance falls into one of four categories: Full (FULL = 100%), High (75% <= HIGH < 100%), Partial (50% <= PARTIAL < 75%), or Poor (POOR < 50%). In addition to assessing the content coverage of the SEKAs within the SE Curriculum, this research also evaluates the explicit inclusion of the main topics of the SEKAs (Kas) as course learning outcomes (CLOs) within the curriculum.

Table XVII: SE undergraduate curriculum Coverage of the SEKAs

SEKA	Coverage	Coverage in the SE Curriculum at JUST														
		SE210	SE220	SE225	SE230	SE310	SE320	SE321	SE324	SE323	SE430	SE431	SE432	SE440	SE441	SE471
Software Requirements	FULL															
Software Design	HIGH	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Software Construction	PARTIAL	√		√	√	√	√	√	√	√	√	√	√	√	√	√
Software Testing	FULL															
Software Maintenance	PARTIAL					√				√						
Software Configuration	PARTIAL															√
SE Management	FULL															√
SE process	PARTIAL				√					√	√					√
SE Models & Methods	HIGH		√						√	√				√	√	
Software Quality	FULL												√			√
SE Professional Practice	POOR														√	
SE Economics	POOR														√	
Computing Foundation	HIGH	In several programming/database/data structure courses.														
Mathematical Foundation	HIGH	Through the MATH241 Discrete Mathematics course														
Engineering Foundation	POOR	Almost none of the topics in this knowledge area are covered in the curriculum.														

The primary outcome of this research indicates that the SE Undergraduate Program's Curriculum at JUST is:

- High-to-Full Compliance: The research findings demonstrate a high-to-full level of compliance with SEKA#1 SW-RQTs, SEKA#2 SW-DS, SEKA#4 SW-TS, SEKA#7 SE Management, SEKA#9 SE Models and Methods, SEKA#10 Software Quality, SEKA#13 CFs, and SEKA#14 Mathematical Foundations within the SE Undergraduate Program's Curriculum at JUST.
- Partial Compliance: The analysis indicates a partial level of compliance with SEKA#3 SW-CN, SEKA#5 SW-MN, SEKA#6 SW-CM, and SEKA#8 SE Process within the SE Undergraduate Program's Curriculum at JUST.
- Poor Compliance: The examination reveals poor compliance with SEKA#11 SE Professional Practice, SEKA#12 SE Economics, and SEKA#15 EFKAs within the SE Undergraduate Program's Curriculum at JUST.

Table XVIII: SE undergraduate curriculum compliance with the SEKAs

SEKA	Full Compliance (100%)	High Compliance (75%-100%)	Partial Compliance (50%-75%)	Poor Compliance (Below 50%)
Software Requirements	✓			
Software Design		✓		
Software Construction			✓	
Software Testing	✓			
Software Maintenance			✓	
Software Configuration Management			✓	
Software Engineering Management	✓			
Software Engineering Process			✓	
Software Engineering Models & Methods		✓		
Software Quality	✓			
Software Engineering Professional Practice				✓
Software Engineering Economics				✓
Computing Foundations		✓		
Mathematical Foundations		✓		
Engineering Foundations				✓

The SE Curriculum at JUST must address partial or poor compliance/coverage. Therefore, the author proposes the following recommendations:

1- Introduce New Courses to Address SEKA Gaps:

- Implement a dedicated course on SW-CN, drawing from the content outlined in chapter 3 of the IEEE-CS SWEBOK.
- Introduce a course focused on SW-MN, incorporating materials from chapter 5 of the IEEE-CS SWEBOK.
- Develop a course covering SW-CM, utilizing resources from chapter 6 of the IEEE-CS SWEBOK.
- Establish a course dedicated to the SE Process, integrating concepts from chapter 8 of the IEEE-CS SWEBOK.
- Create a course centered on SE Professional Practice, incorporating principles from chapter 11 of the IEEE-CS SWEBOK.
- Introduce a course on SE Economics, drawing from content outlined in the chapter of the IEEE-CS SWEBOK.
- Implement a course covering EFKAs, utilizing materials from chapter 15 of the IEEE-CS SWEBOK.

2- The SE441 Software Quality Assurance course is a core rather than an elective course.

6. Conclusions

This paper evaluates the compliance of the SE Undergraduate Program at JUST with the fifteen SEKAs outlined in the IEEE-CS SE Body of Knowledge (SWEBOK). SEKA#1 SW-RQTs, SEKA#2 SW-DS, SEKA#4 SW-TS, SEKA#7 SE Management, SEKA#9 SE Models and Methods, SEKA#10 Software Quality, SEKA#13 CFs, and SEKA#14 Mathematical Foundations were found to be fully or highly covered. However, SEKA#3 SW-CN, SEKA#5 SW-MN, SEKA#6 SW-CM,

and SEKA#8 SE Process were only partially covered. Moreover, SEKA#11 SE Professional Practice, SEKA#12 SE Economics, and SEKA#15 EFKAs needed better coverage within the curriculum. To address these deficiencies, the author recommends adding specialized courses to cover the partially or poorly covered SEKAs comprehensively. These include SEKA#3 SW-CN, SEKA#5 SW-MN, SEKA#6 SW-CM, SEKA#8 SE Process, SEKA#11 SE Professional Practice, SEKA#12 SE Economics, and SEKA#15 EFKAs

References

[Note: All Hyperlinks last seen on February 22, 2024]

Sites References

- [1] ABET Accreditation; <https://www.abet.org/accreditation/>
- [2] JUST Website: www.just.edu.jo
- [3] IEEE Computer Society; <https://www.computer.org/>
- [4] Institute of Engineering and Technology Site – IET Accreditation. <https://www.theiet.org/career/accreditation/academic-accreditation/>

Literature Survey References

- [5] Barzilay, O., et. al., "A multi-dimensional SE course." IEEE Transactions on Education 52, no. 3 (2009): 413-424.
- [6] Bastarrica, M. et al., "What can students get from a SE capstone course?" In 2017 IEEE/ACM 39th International Conference on Software Engineering: SE Education and Training Track (ICSESEET), pp. 137-145. IEEE, 2017.
- [7] Daimi, K., "Strengthening elements of teamwork, innovation, and creativity in a SE program." The Journal of Engineering Entrepreneurship 3, no. 1 (2012): 35-50.
- [8] Dekhane, S. et al., "Work in progress—Interdisciplinary collaboration for a meaningful experience in a Software development course." In 2010 IEEE Frontiers in Education Conference (FIE), pp. S1D-1. IEEE, 2010.
- [9] Fox, A., et. al. (2014). SE curriculum technology transfer: lessons learned from MOOCs and SPOCs. UC Berkeley EECS Technical Report.
- [10] Garousi, V. et al., Challenges and best practices in industry-academia collaborations in Software Engineering: A systematic literature review. Information and Software Technology 79 (2016): 106-127.

- [11] Garousi, V. et al., "Closing the gap between SE education and industrial needs." IEEE Software (2019).
- [12] Meziane, F., et al., "A comparison of computer science and SE programs in English universities". The 17th Conference on SE Education and Training, 2004. Proceedings. (pp. 65-70). IEEE.
- [13] ISO/IEC TR 19759:2015 Software Engineering: Guide to the SE body of knowledge (SWEBOK). ISO/IEC TR 19759:2015 - SE — Guide to the SE body of knowledge (SWEBOK)
- [14] Radaideh, M., "Benchmarking the SE undergraduate program curriculum at Jordan University of Science and Technology with the IEEE SE body of knowledge: (SEKAs #11-15)". The 2021 International Conference on Computational Science and Computational Intelligence (CSCI'21: December 15-17, 2021; Las Vegas, USA) <https://www.american-cse.org/csci2021/>. Published by the IEEE CPS - Benchmarking the SE undergraduate program curriculum at Jordan University of Science and Technology with the IEEE SE body of knowledge: (SEKAs #11-15) | IEEE Conference Publication | IEEE Xplore.
- [15] Radaideh, M., "Shifting the Paradigms from Teaching Project Management to Teaching Software Project Management at the Jordan University of Science and Technology According to the IEEE SW-MGKnowledge Area". The 2021 International Conference on Computational Science and Computational Intelligence (CSCI'21: December 15-17, 2021; Las Vegas, USA) <https://www.american-cse.org/csci2021/>. Published by the IEEE CPS - Shifting the paradigms from teaching project management to teaching software project management at Jordan University of Science and Technology based on the IEEE SW-MGknowledge area | IEEE Conference Publication | IEEE Xplore.
- [16] Radaideh, M., "Benchmarking the SE Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE SE Body of Knowledge (SEKAs #1-5)". Published in the SPRINGER NATURE Research Book Series: Advances in Software Engineering, Education, and e-Learning, pp 747-768, September 2021 (https://link.springer.com/chapter/10.1007/978-3-030-70873-3_53).
- [17] Radaideh, M., "Benchmarking the SE Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE SE Body of Knowledge (SWE Knowledge Areas #6-10)". Published in the SPRINGER NATURE Research Book Series: Advances in Software Engineering, Education, and e-Learning, pp 85-100, September 2021 (https://link.springer.com/chapter/10.1007/978-3-030-70873-3_7).
- [18] Qiu, M. et al., "A problem-based learning approach to teaching an advanced SE course." 2010, the Second International Workshop on Education Technology and Computer Science was held. 2010.

SEKA#1 SW-RQTs - Reference Standards

- [19] ISO/IEC TR 19759:2015 SE - Guide to the SE body of the knowledge (SWEBOK). <https://www.iso.org/standard/67604.html>
- [20] ISO/IEC/IEEE 24765:2017 Systems and Software Engineering-Vocabulary. <https://www.iso.org/standard/71952.html>
- [21] ISO/IEC 19761:2011 Software Engineering-COSMIC: A Functional Size Measurement Method. <https://www.iso.org/standard/54849.html>
- [22] ISO/IEC 20926:2009 Software and Systems Engineering - Software Measurement - IFPUG Functional Size Measurement Method. <https://www.iso.org/standard/51717.html>
- [23] ISO/IEC 20968:2002 SE -Mk II Function Point Analysis - Counting Practices Manual. <https://www.iso.org/standard/35603.html>
- [24] ISO/IEC 24570:2005 SE - NESMA Functional Size Measurement Method Version 2.1 - Definitions and Counting Guidelines for applying Function Point Analysis. <https://www.iso.org/standard/37289.html>
- [25] ISO/IEC 19501:2005 Information Technology - Open Distributed Processing - Unified Modelling Language (UML) Version 1.4.2. <https://www.iso.org/standard/32620.html>
- [26] ISO/IEC 19505:2012 [two parts] Information Technology - Object Management Group Unified Modelling Language (OMG UML). Part I: <https://www.iso.org/standard/32624.html> Part II: <https://www.iso.org/standard/52854.html>

SEKA#2 SW-DS - Reference Standards

- [27] ISO/IEC/IEEE 42010:2022 Software, Systems, and Enterprise - Architecture Description. <https://www.iso.org/standard/74393.html>
- [28] IEEE Std. 1016-2009 Standard for Information Technology - Systems Design - SW-DS Descriptions. <https://ieeexplore.ieee.org/document/5167255>

- [29] ISO/IEC/IEEE 26514:2022 Systems and SE - Design and Development of Information for Users. <https://www.iso.org/standard/77451.html>

SEKA#3 Software Construction - Reference Standards

- [30] ISO/IEC TR 24772:2013 Information Technology - Programming Languages -Guidance to Avoiding Vulnerabilities in Programming Languages through Language Selection and Use. <https://www.iso.org/standard/61457.html>
- [31] IEEE Std. 1008-1987 Standard for Software Unit Testing. <https://ieeexplore.ieee.org/document/27763>
- [32] ISO/IEC/IEEE 29119-4:2021 Software and Systems Engineering - SW-TS - Part 4: Test Techniques. <https://www.iso.org/standard/79430.html>
- [33] ISO/IEC/IEEE 26515:2018 Systems and SE - Developing Information for Users in an Agile Environment. <https://ieeexplore.ieee.org/document/8584455>
- [34] IEEE Std. 1517-2010 Standard for Information Technology - System and Software Life Cycle Processes - Reuse Process. <https://ieeexplore.ieee.org/document/5551093>

SEKA#4 SW-TS - Reference Standards

- [35] IEEE Std. 829-2008 Standard for Software and System Test Documentation. <https://ieeexplore.ieee.org/document/4578383>
- [36] IEEE Std. 1008-1987 Standard for Software Unit Testing. <https://ieeexplore.ieee.org/document/27763>
- [37] ISO/IEC/IEEE 29119-4:2021 Software and Systems Engineering - SW-TS - Part 4: Test Techniques. <https://www.iso.org/standard/79430.html>
- [38] ISO/IEC/IEEE 26513-2017 International Standard - Systems and SE - Requirements for Testers and Reviewers of Information for Users. <https://ieeexplore.ieee.org/document/8085435>
- [39] IEEE Std. 1012-2016 Standard for System, Software, and Hardware Verification and Validation. <https://ieeexplore.ieee.org/document/8055462>
- [40] IEEE Std. 1044-2009 Standard for Classification for Software Anomalies. <https://ieeexplore.ieee.org/document/5399061>

SEKA#5 Software Configuration Management - Reference Standards

- [41] ISO/IEC/IEEE 14764:2022 SE - Software Life Cycle Processes - Maintenance. <https://www.iso.org/standard/80710.html>

SEKA#6 Software Configuration Management - Reference Standards

- [42] IEEE Std. 828-2012 Standard for Configuration Management in Systems and Software Engineering. <https://ieeexplore.ieee.org/document/6170935>

SEKA#7 SW-MG- Reference Standards

- [43] ISO/IEC/IEEE 16326:2019 Systems and SE - Life Cycle Processes - Project Management. <https://www.iso.org/standard/75276.html>
- [44] ISO/IEC/IEEE 16085:2021 System and SE - Life Cycle Processes - Risk Management. <https://www.iso.org/standard/74371.html>
- [45] ISO/IEC/IEEE 15939:2017 System and SE - Measurement Process. <https://www.iso.org/standard/71197.html>
- [46] IEEE Std. 1062-1998 Recommended Practice for Software Acquisition. <https://ieeexplore.ieee.org/document/741938>
- [47] ISO/IEC/IEEE 26512:2018 Systems and SE - Requirements for Acquirers and Suppliers of Information for Users. <https://www.iso.org/standard/72088.html>
- [48] IEEE Std. 1028-2008 Standard for Software Reviews and Audits. <https://ieeexplore.ieee.org/document/4601584>
- [49] IEEE Std. 1061-1998 Standard for Software Quality Metrics Methodology. <https://ieeexplore.ieee.org/document/749159>
- [50] ISO/IEC/IEEE 26515:2018 Systems and SE - Developing Information for Users in an Agile Environment. <https://ieeexplore.ieee.org/document/8584455>

SEKA#8 SE Process - Reference Standards

- [51] ISO/IEC/IEEE 15289:2019 Systems and SE - Content of Lifecycle Information Items (Documentation). <https://www.iso.org/standard/74909.html> Link:
- [52] ISO/IEC/IEEE 24748-2:2018 Systems and SE - Life Cycle Management - Part 2: Guidelines for applying ISO/IEC/IEEE 15288 (System Life Cycle Processes). <https://www.iso.org/standard/84661.html>
- [53] ISO/IEC/IEEE 24748-3:2020 Systems and SE - Life Cycle Management - Part 3: Guidelines for applying ISO/IEC/IEEE 12207 (Software Life Cycle Processes). <https://www.iso.org/standard/77698.html>

- [54] ISO/IEC/IEEE 24748-1:2018 Systems and SE - Life Cycle Management - Part 1: Guidelines for Life Cycle Management. <https://www.iso.org/standard/84709.html>
- [55] IEEE Std. 1517-2010 Standard for Information Technology - System and Software Life Cycle Processes - Reuse Processes. <https://ieeexplore.ieee.org/document/5551093>
- [56] ISO/IEC/IEEE 24774:2021 Systems and SE - Life Cycle Management - Specification for Process Description. <https://www.iso.org/standard/78981.html>
- [57] ISO/IEC TR 29110-5-1-2:2011 SE - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and Engineering Guide: Generic Profile Group: Basic Profile. <https://www.iso.org/standard/51153.html>
- [58] IEEE Std. 1074-2006 Standard for Developing a Software Project Life Cycle Process. <https://ieeexplore.ieee.org/document/1665059>
- [59] IEEE Std. 828-2012 Standard for Configuration Management in Systems and Software Engineering. <https://ieeexplore.ieee.org/document/6170935>
- [60] ISO/IEC/IEEE 14764:2022 SE - Software Life Cycle Processes - Maintenance. <https://www.iso.org/standard/80710.html>
- [61] ISO/IEC/IEEE 15026-4:2021 Systems and SE - Systems and Software Assurance - Part 4: Assurance in the Life Cycle. Link: <https://www.iso.org/standard/74396.html>
- [62] ISO/IEC/IEEE 15939:2017 System and SE - Measurement Process. <https://www.iso.org/standard/71197.html>
- [63] ISO/IEC 15940:2013 Systems and SE - SE Environment Services. <https://www.iso.org/standard/60100.html>
- [64] ISO/IEC/IEEE 16085:2021 System and SE - Life Cycle Processes - Risk Management. <https://www.iso.org/standard/74371.html>
- [65] ISO/IEC/IEEE 16326:2019 Systems and SE Lifecycle Processes -Project Management. <https://www.iso.org/standard/75276.html>
- [66] ISO/IEC/IEEE 29148:2018 Systems and SE - System Life Cycle Processes - Requirements Engineering. Link: <https://www.iso.org/standard/72089.html>
- [67] ISO/IEC 20000-1:2018 Information Technology - Service Management-Part 1: Service Management System Requirements. <https://www.iso.org/standard/70636.html>
- SEKA#9 SE Models and Methods - Reference Standards**
- [68] ISO/IEC/IEEE 26515:2018 Systems and SE - Developing Information for Users in an Agile Environment. <https://ieeexplore.ieee.org/document/8584455>
- [69] IEEE Std. 1320.1-1998 Standard for Functional Modelling Language-Syntax and Semantics for IDEF0. <https://ieeexplore.ieee.org/document/749110>
- [70] 31320-2-2012-ISO/IEC/IEEE International Standard- Information technology - Modelling Languages - Part 2: Syntax and Semantics For IDEF1X97(IDEFObject). <https://ieeexplore.ieee.org/document/6357338>
- [71] ISO/IEC 19501:2005 Information Technology - Open Distributed Processing-Unified Modelling Language (UML) Version 1.4.2. <https://www.iso.org/standard/32620.html>
- [72] ISO/IEC 19505:2012 [two parts] Information Technology - Object Management Group Unified Modelling Language (OMG UML). Part I: <https://www.iso.org/standard/32624.html> Part II: <https://www.iso.org/standard/52854.html>
- [73] ISO/IEC 19506:2012 Information Technology - Object Management Group Architecture - Driven Modernization (ADM) - Knowledge Discovery Meta-Model (KDM). <https://www.iso.org/standard/32625.html>
- [74] ISO/IEC 19507:2012 Information Technology - Object Management Group Object Constraint Language (OCL). <https://www.iso.org/standard/57306.html>
- [75] ISO/IEC 15940:2006 Information Technology - SE Environment Services. <https://www.iso.org/standard/29573.html>
- [76] IEEE Std. 14102-2010 Standard Adoption of ISO/IEC 14102:2008 Information Technology - Guideline for Evaluating and Selection CASE Tools. <https://ieeexplore.ieee.org/document/5567083>
- [77] IEEE Std. 1175.1-2002 Guide for CASE Tool Interconnections - Classification and Description. <https://ieeexplore.ieee.org/document/1178882>
- [78] IEEE Std. 1175.2-2006 Recommended Practice for CASE Tool Interconnection-Characterization of Interconnections. <https://ieeexplore.ieee.org/document/4068343>
- [79] IEEE Std. 1175.3-2004 Standard for CASE Tool Interconnections - Reference Model for Specifying

- Software Behaviour.
<https://ieeexplore.ieee.org/document/1350464>
- [80] IEEE Std. 1175.4-2008 Standard for CASE Tool Interconnections - Reference Model for Specifying System Behaviour.
<https://ieeexplore.ieee.org/document/4839189>
- SEKA#10 Software Quality - Reference Standards**
- [81] ISO/ IEC/IEEE 90003-2018 International Standard- SE - Guidelines for applying ISO 9001:2015 to Computer Software.
<https://ieeexplore.ieee.org/document/8559961>
- [82] IEEE Std. 730-2014 Standard for Software Quality Assurance Processes.
<https://ieeexplore.ieee.org/document/6835311>
- [83] ISO/IEC 25000:2014 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-Guide to SQuaRE. <https://www.iso.org/standard/64764.html>
- [84] ISO/IEC 25010:2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models. Link: <https://www.iso.org/standard/35733.html>
- [85] ISO/ TR 25060:2023 Systems and SE - Systems and Software Quality Requirements and Evaluation (SQuaRE) - General Framework for Common Industry Format (CIF) for Usability - related Information.
<https://www.iso.org/standard/83763.html>
- [86] IEEE Std. 982.1-2005 Standard for Dictionary of Measures of the Software Aspects of Dependability.
<https://ieeexplore.ieee.org/document/1634994>
- [87] IEEE Std. 1633-2016 Recommended Practice for Software Reliability.
<https://ieeexplore.ieee.org/document/7827907>
- [88] IEEE Std. 1061-1998 Standard for Software Quality Metrics Methodology.
<https://ieeexplore.ieee.org/document/749159>
- [89] IEEE Std. 1012-2016 Standard for System, Software, and Hardware Verification and Validation.
<https://ieeexplore.ieee.org/document/8055462>
- [90] ISO/IEC/IEEE 15026-1:2019 Systems and SE - Systems and Software Assurance - Part 1: Concepts and Vocabulary.
<https://www.iso.org/standard/73567.html>
- [91] ISO/IEC/IEEE 15026-2:2022 Systems and SE - Systems and Software Assurance - Part 2: Assurance Case. <https://www.iso.org/standard/80625.html>
- [92] ISO/IEC/IEEE DIS 15026-3 Systems and Software Engineering-Systems and Software Assurance - Part 3: System Integrity Levels.
<https://www.iso.org/standard/84444.html>
- [93] ISO/IEC/IEEE 15026-4:2021 Systems and Software Engineering-Systems and Software Assurance - Part 4: Assurance in the Life Cycle.
<https://www.iso.org/standard/74396.html>
- [94] IEEE Std. 1228-1994 Standard for Software Safety Plans. <https://ieeexplore.ieee.org/document/9097571>
- [95] IEEE Std. 829-2008 Standard for Software and System Test Documentation.
<https://ieeexplore.ieee.org/document/4578383>
- [96] IEEE Std. 1008-1987 Standard for Software Unit Testing. <https://ieeexplore.ieee.org/document/27763>
- [97] ISO/IEC/IEEE 26513-2017 International Standard - Systems and SE - Requirements for Testers and Reviewers of Information for Users.
<https://ieeexplore.ieee.org/document/8085435>
- [98] ISO/IEC/IEEE 29119-4:2021 Software and Systems Engineering - SW-TS - Part 4: Test Techniques.
<https://www.iso.org/standard/79430.html>
- SEKA#11 SE Professional Practice - Reference Standards**
- [99] ISO/IEC TR 19759:2015 Software Engineering-Guide to the SE Body Knowledge (SWEBOK).
<https://www.iso.org/standard/67604.html>
- [100] ISO/IEC 24773-1:2019 Software and Systems Engineering - Certification of Software and Systems Engineering Professionals - Part 1: General Requirements.
<https://www.iso.org/standard/69724.html>