

Index

A

- Absolute conformance, software
 - engineering process standards, 208–209
- Abstraction technology, software design, 102
- Acceptance tests:
 - acquisition process, 296
 - software requirements, 93
 - software testing, 127
- Accounting, software configuration management, 159–160
- Acquisition process, life cycle framework, 296–298
 - systems agreement, 344
- Activities planning:
 - software maintenance, 143–145
 - software testing, 132–133
- Adaptation, software engineering process, 194
- Adaptive software maintenance, 140
- Adherence to process, for software engineering standards, 10
- Agile methods, software construction, 116
- Agreement process, systems life cycle framework, 344
- AIAA. *See* American Institute of Aeronautics and Astronautics
 - AIAA G-010-1993 software standard, 31–32
 - AIAA R-013-1992 software standard, 31–32
- Alpha/beta testing, software testing, 128
- Alternative conformance, software engineering process standards, 209
- American Institute of Aeronautics and Astronautics (AIAA):
 - ANSI and, 30
 - IEEE standards and, 35
 - structure and function, 30–32
- American National Standards Institute (ANSI), structure and function, 29–30
- American Society for Testing and Materials (ASTM), software engineering standards, 14
- Analysis of software requirements, 84–88
 - architectural design, 85–87
 - classification, 84–85
 - conceptual modeling, 85
 - high-integrity systems, 88
 - life cycle processes framework, 303–305
 - negotiation, 87

- Analytical techniques, software quality management, 238
 - Analytic paradigm, software engineering process measurement, 197–198
 - Analyzability measures, software maintenance, 145
 - Anomaly data:
 - defect characterization, 238
 - problem resolution framework, 323–324
 - ANSI. *See* American National Standards Institute
 - Application guides:
 - organization of, 67
 - software quality requirements, 237
 - Application program interfaces (APIs), software construction, 116
 - Applications analysis, software testing, 130
 - Architectural coherence, software engineering standards for, 58–62
 - consistency, 59
 - process abstraction model, 61–62
 - product quality model, 59–60
 - Architectural design:
 - life cycle process framework, 302–303
 - software requirements allocation and, 85–87
 - systems engineering, 265–267
 - life cycle framework, 350–351
 - Architectural style, software design, 105
 - Architecture requirements, software design and, 103–105
 - Assessment, software engineering process, 194–195
 - Asset management process, life cycle framework, 325–327
 - Auditing:
 - life cycle framework, 323
 - quality management, ISO 19011:2002 guidelines, 271–272
 - software configuration management, 160–163
 - software quality, 236–237
 - Automation, software engineering process, 194
- B**
- Back-to-back testing, software testing, 128
 - Baseline mechanism, software configuration management, 156–157
 - Base measures, software engineering process standards, 213–214
 - Basic interoperability data model (BIDM):
 - IEEE Std 1420.1-1995, 222
 - standard for, 227
 - Basili model:
 - process abstraction, 288–291
 - software engineering standards, 61–62
 - implementation guidelines, 190
 - process-oriented design, 284
 - Behavioral theory, software design notation, 107–108
 - Benchmarking paradigm, software engineering process measurement, 197–198
 - “Best practices,” in software engineering, 7–8
 - “Big bang” test, software testing, 127
 - Body of knowledge:
 - overview of SWEBOCK contents, 73–76
 - software engineering, 5–6
 - Bottom-up approach, software testing, 127
 - Boundary-value analysis, specification-based software testing, 129
 - “Breaking the program” technique, software testing, 126, 128
 - Business protection, software engineering standards, 14
- C**
- Capability Maturity Model Integration (CMMI):
 - improvement process, 331–332
 - life cycle framework, 331
 - software engineering process, 194–195
 - Case generation, software testing, 133
 - CASE tools:
 - evaluation, 224, 227–228, 239
 - integration of, 223, 225
 - CEN. *See* European Committee for Standardization
 - CENELEC. *See* European Committee for Electrotechnical Standardization
 - Changeability measures, software maintenance, 145
 - Change management:
 - software construction, 115
 - software engineering process, 185–189
 - infrastructure, 186

- models, 187–188
- organizational environment, 188
- plan-do-check-act cycle, 187
- training issues, 189
- Change requests, software configuration management, 159
- Characterization of bounds, in SWEBOK guide, 74
- Classification, of software requirements, 84–85
- Closure, software project management, 176
- CMM-Based Appraisal for Internal Process Improvement (CBA-IPI), software engineering process assessment, 194–195
- Code-based software testing, 129
- Code generators, software construction with, 221
- Collection guides, standards organization, 66
- Communication framework, software engineering standards, 13
- Community of interest, for software engineering standards, 10
- Compilers, software construction with, 221
- Completeness, software design, 102
- Completion, acquisition process, 296
- Complexity minimization, software construction, 115
- Compliance, with software engineering standards, 50–51
- Component distribution, software design, 103
- “Component factory” model:
 - process abstraction, 288–291
 - software engineering process, 190
- Comprehension tools, software maintenance, 221
- Computer engineering, IEEE curricula for, 250–251
- Computer science, knowledge areas, 251
- Computer Society Software and Systems Engineering Committee (IEEE), 38–42
 - history, 39
 - integration with ISO/IEC JTC 1/SC 7, 41–42
 - organization, 39–41
- Concept of Operations document:
 - software acquisition, 297
 - stakeholder requirements, 346–347
 - standard for, 257
 - system definition, 88
 - systems engineering, 264–265
- Conceptual level process standards:
 - process abstraction, 289
 - software engineering processes, 192–193
- Conceptual modeling, software requirements, 85
- Conceptual view, of software engineering process, 190
- Concurrency, software design, 103
- Configuration languages, software construction, 119
- Configuration management (CM). *See also* Software configuration management
 - life cycle process framework, 317–318
 - systems life cycle processes, 345
 - software maintenance, 145
 - software project management, process control, 174–175
 - tools for, 221–222
- Configuration testing, 128
- Conformance:
 - software engineering standards and, 50–51, 207–208
 - software testing, 128
- Consensual validation, in SWEBOK guide, 74
- Consensus principles, software engineering standards, 10
- Consistency, architectural coherence in software engineering and, 59
- Constraints, software configuration management, 153
- Consumer protection, software engineering standards, 13
- Content compliance, software project management standard, 178
- Context, for software design, 101
- Continuous capability model, software engineering process, 194–195
- Contract preparation and update, acquisition process, 296
- Control:
 - software configuration management, 158–159
 - software design, 103
- Control flow testing, code-based software evaluation, 129

- Corrective software maintenance, 140
- Cost estimation:
 - software maintenance, 141–142
 - software project management, 171
 - software quality, 232
- Coupling and cohesion, software design, 102
- Coverage measurement, software testing evaluation, 131
- Cross-project framework:
 - life cycle processes, 295, 334–335
 - domain (reuse) engineering, 334–335
 - software engineering standards, 52
- D**
- Data flow testing, code-based software evaluation, 129
- Data framework, software engineering standards, 55–58
- Data persistence, software design, 103
- Debuggers, software construction with, 221
- Decision tables, specification-based software testing, 129
- Decomposition and modularization, software design, 102
- Defect tracking:
 - software quality management, 238
 - software testing, 133
 - tools for, 222
- Deliverables, software project management, 170
- Delivery, software configuration management, 162–163
- Department of Defense (DoD), life cycle standards, history and concepts, 283–285
- Derived measures, software engineering process standards, 213–214
- Description:
 - data framework for software standards, 56–58
 - system requirements, 89
- Design concepts:
 - life cycle framework, detailed software design, 306–307
 - patterns, in software design, 105
 - software construction, 118
 - software design, 101
 - knowledge area, 99–111
 - fundamentals, 101–102
 - key issues, 102–103
 - notations, 107–108
 - quality analysis and evaluation, 105–107
 - scope, 99–100
 - standards, 109–111
 - strategies and methods, 108–109
 - structure and architecture, 103–105
 - summary, 100–101
 - SWEBOK guidelines on, 74
 - tools for, 220
- Deterministic testing, software evaluation, 130
- Development process, life cycle framework, 299–313
 - acceptance support, 311
 - analysis activity, 300–302
 - architectural design activity, 302–303
 - coding and testing activity, 307–308
 - implementation activity, 300
 - installation, 310
 - integration, 308–309
 - qualification testing, 309–310
 - software architecture design, 305–306
 - software detailed design, 306–307
 - software requirements analysis, 303–305
 - system integration, 310
 - system qualification testing, 310
- Disposal process, systems life cycle, 353
- Documentation:
 - life cycle processes standards, 314–316
 - software construction:
 - testing, 119
 - user documentation standards, 120–121
 - software project management, 170
 - software requirements, 81–82
 - software testing process, 132
 - standards, organization of, 64
- Document Item Description (DIDs), 315
- Domain engineering, cross-project processes, 334–335
- Domain knowledge, software requirement elicitation, 84
- Domain-specific languages, software construction, 119
- Dynamic verification:
 - software quality management, 238
 - software testing, 125–126

- E**
- Efficiency, in product quality model, 60
 - Effort estimation, software project management, 171
 - Egoless programming, software testing process, 132
 - EIA. *See* Electronic Industries Alliance
 - EIA-632 standard, 33
 - EIA 649:1998 *National Consensus Standard for Configuration Management*, 151
 - EIA-649 standard, 33
 - EIA-731 standard, 33–34
 - Electronic Industries Alliance (EIA). *See also* IEEE/EIA 12207.1-1996
 - ANSI and, 30
 - structure and function, 32–34
 - Element standards, organization of, 67
 - Elicitation, software requirements, 84
 - Enabling techniques:
 - software design, 102
 - systems engineering, life cycle framework, 339
 - Enactment phase, software project management, 173
 - Engineering:
 - computer engineering curricula, 250–251
 - software engineering as, 3–5
 - Enhancement tools, software configuration with, 222
 - Enterprise process, systems life cycle, 344–345
 - Environmental issues:
 - software testing, 133
 - systems life cycle, enterprise process, 344
 - Equivalence partitioning, specification-based software testing, 129
 - Ergonomics:
 - ISO TC 159 standard, 26
 - software ergonomics, 255–256
 - Error guessing, fault-based software testing, 129
 - Error handling, software design, 103
 - Estimation techniques, software testing process, 132
 - European Committee for Electrotechnical Standardization (CENELEC), structure and function, 28
 - European Committee for Standardization (CEN), structure and function, 28
 - Evaluation criteria:
 - software engineering standards, 13
 - process framework for, 54–55
 - software project management, 175
 - software testing, 131
 - results evaluation, 133
 - tool evaluation, 223–224
 - Events management, software design, 103
 - Exception handling, software design, 103
 - Execution protocols, software testing, 133
 - Experience-based software testing, 128
 - Experience Factory, software engineering process infrastructure, 186
 - Experimental studies, software engineering process measurement, 198
 - External quality:
 - ISO/IEC 9126-1:2001 standard for, 96
 - software products, 233
 - Extreme programming, software construction, 116
- F**
- Failure-causing inputs, software testing, 126
 - Fault-based software testing, 129
 - Fault classification, software testing, 130
 - Fault density, software testing, 131
 - Fault seeding, software testing evaluation, 131
 - Federal Information Processing Standard Publication 183 (FIPS Pub. 183), 226
 - Finite-state-machine-based evaluation, specification-based software testing, 129
 - Formal engineering methods, 224
 - Formal specifications, specification-based software testing, 129
 - Format compliance, software project management standard, 178
 - Frameworks, software design, 105
 - Functional Configuration Audit (FCA), software configuration management, 161–162
 - Functionality, in product quality model, 60

Functional requirements, defined, 80

Functional size measurement:

IEEE Std 14143.1-2000 standard,
96–97

software maintenance, 141–142

software requirements, 94

Functional testing, software evaluation,
130

G

Goals identification, software
requirements, 84

Government Electronics and Information
Technology Association (GEIA),
standards development, 33–34

*Guide for the Application of ISO/IEC
12207 to Project Management*,
170

Guidelines, software testing process, 132

*Guidelines for quality and/or
environmental management
systems auditing*, 271–272

*Guidelines for Quality Management in
Projects*, 170

Guides (IEC), 23–25

*Guide to the Project Management Body
of Knowledge, The (PMBOK
Guide)*, 252

life cycle framework, 329

systems engineering PMI standard,
267–268

*Guide to the Software Engineering Body
of Knowledge (SWEBOK)*:

knowledge areas in, 74–76

mathematics, 251

purpose of, 73–74

software configuration management,
149–150

software construction, 113–115

software design, 99–101

software engineering management,
165–167

software engineering standards, 46–48

software maintenance, 137–138

software testing scope and summary,
123–125

standards organization in, 69

H

Heuristic engineering methods, 224

Hierarchical techniques, software
testing, 127

High-integrity systems, software
requirements, 88

I

IDEF modeling, software requirements,
85

IDEF1X97 modeling, 226

life cycle process framework, 301–302

IDEF0 modeling:

IEEE Std 1320.1-1998 standard,
225–226

life cycle process framework, 301–302
requirements modeling tools, 220

software design notation, 108

software engineering process, 193

Identification, software configuration
management, 156

IEC. *See* International Electrotechnical
Commission

IEC 880 standard, commercial life cycle
processes, 286–288

IEC SC 65A standard, 25

S2ESC coordination with, 48

IEC TC 56 Dependability Standard,
24–25

S2ESC coordination with, 48

IEC TC 62 Electrical Equipment in
Medical Practice standard, 25

IEEE. *See* Institute of Electrical and
Electronics Engineers

IEEE/EIA 12207-1996 standard:

architectural design, 86–87

Basili framework, 61–62

Clause 5.1, Acquisition process, 298

Clause 5.2, Supply process, software
engineering training, 189

Clause 5.3, Development process,
90–91, 106

software construction, 118

Clause 5.3.4.3, software configuration
management, 156–157

Clause 5.3.9.5, software configuration
management, 156–157

Clause 5.3.11.4, software configuration
management, 156–157

Clause 5.4, software maintenance, 144

Clause 6.1.1.1, software project
documentation, 170

Clause 6.2.2.1, software configuration
management, 156

Clause 6.2.3.1, software configuration
management, 158

- Clause 6.2.4.1, software configuration management, 160
- Clause 6.2.5.1, software configuration management, 161
- Clause 6.2.6.1, software configuration management and delivery, 162–163
- Clause 6.3.4, quality management, 319
- Clause 6.4, Development process, 92
- Clause 6.4, Verification process, 106
- Clause 6.5, Validation process, 92
- collection coordination with S2ESC, 47–48
- data framework, 55–58
- life cycle framework:
 - acquisition, 296–298
 - basic principles, 293–294, 365
 - development process, 299–313
 - maintenance process, 311–313
 - primary processes, 295–298
 - software integration, 308–309
 - software qualification testing, 309–310
 - software requirements analysis, 303–305
 - supply framework, 298–299
- organizationally adopted processes, 50–51
- reference process framework, 291–292
- software configuration management, 151–152
- software construction, 116–117
 - integration requirements, 120
- software design process, 102
- software engineering management:
 - closure phase, 176
 - enactment phase, 173
 - initiation and scope definition, 168–169
 - project planning, 169
 - review and evaluation, 175
- software engineering process:
 - conceptual level process standards, 192–193
 - conformance, 207–208
 - improvement standards, 188
 - infrastructure standards, 186
 - ISO/IEC 12207:1995
 - implementation, 204–206
 - ISO/IEC 12207 *vs.*, 209–210
 - life cycle data guidelines, 206–207
 - life cycle processes, 201–202
 - reference-level process, 191–192
 - reuse process, 199
 - software maintenance, 139
 - process models, 142–143
 - software quality management and assurance, 234–235
 - software requirements process, 82–84
 - software review, 93
 - software structure and architecture, 104
 - software testing, 133–134
 - system-software integration, 49
 - life cycle framework, 341–344
 - uniform process framework, 51–58
- IEEE/EIA 12207.1-1996:
 - data framework, 55–58, 365
 - documentation standards, 315–316
 - software project management, 171
 - software requirement description, 82
 - system requirements specification, 89
- IEEE/EIA 12207.2-1997, software engineering process, implementation considerations, 207, 365
- IEEE Std 610.12-1990:
 - software configuration management, 150–151
 - software engineering terminology, 277–278, 359
- IEEE Std 730-2002:
 - development of, 40–41
 - software configuration management, 153
 - software engineering process, 205
 - software project management, 172
 - software quality assurance, 235, 240–242, 359
 - life cycle processes framework, 318–320
- IEEE Std 828-1998:
 - configuration management, 317–318
 - software configuration management, 153–154, 359
 - accounting, 160
 - auditing, 161–162
 - basic principles, 163
 - control guidelines, 158–159
 - planning guidelines, 157
 - software project management, 175

- IEEE Std 829-1998:
 - software construction testing, 119
 - software integration framework, 308–309
 - software testing documentation, 134–135, 360
 - system integration, life cycle framework, 310
 - system qualification testing, 310
- IEEE Std 830-1998, 58
 - software requirements specification, 90–91, 94–95, 360
 - life cycle framework, 305–306
- IEEE Std 982.1-1988:
 - software quality measurement, 240
 - reliability standards, 242
 - software reliability testing, 131, 360
 - life cycle framework, 325
- IEEE Std 1008-1987:
 - software construction testing, 119
 - software engineering process, 192
 - software unit testing, 127, 136, 360
 - life cycle framework, 307–308
- IEEE Std 1012-2004, software verification and validation, 236, 242–244, 360
 - life cycle process framework, 321–322
- IEEE Std 1016-1998:
 - software design descriptions, 109–110, 360
 - life cycle framework, 306–307
 - software structure and architecture, 104–105
- IEEE Std 1028-1997:
 - software configuration management, auditing, 160–163, 162
 - software construction quality, 119
 - software design quality, 106
 - software project management, evaluation and review, 175
 - software quality reviews and audits, 236–237, 244–245
 - software review, 93, 360
 - auditing process, 323
 - joint review process, 322–323
- IEEE Std 1044-1993:
 - defect characterization, 238
 - software anomalies, 324, 361
 - software engineering process defects, 198
 - software fault classification, 131
 - software project management, process control, 174–175
 - software quality anomalies, 245
- IEEE Std 1045-1992:
 - software productivity metrics, 325, 361
 - software project management:
 - process monitoring, 174
 - software productivity metrics, 177
- IEEE Std 1058-1998, software project management, 170, 361
 - life cycle framework, 329
 - outline, 179–180
 - plans and documentation, 178
- IEEE Std 1061-1998 (R2004):
 - measurement processes, 325
 - quality metrics methodology, 319, 361
 - software engineering process measurement, 197
 - software quality measurement, 240
 - metrics methodology, 245–246
- IEEE Std 1062-1998:
 - recommended practice for software acquisition, 178
 - software acquisition, 296–297, 361
 - software project management, supplier contracts, 173
 - software requirements process, 83–84
- IEEE Std 1063-2001:
 - documentation standards, 316, 361
 - life cycle framework, software user documentation, 306
 - software construction:
 - integration and, 120
 - user documentation, 120–121
- IEEE Std 1074-1997:
 - commercial life cycle processes, 286–288
 - life cycle process, 361
 - implementation activity, 300
 - software engineering process, 193
 - life cycle processes, 198–201
- IEEE Std 1175.1-2002:
 - CASE tools integration, 223, 225, 362
 - infrastructural management framework, 330
- IEEE Std 1219-1998:
 - software engineering process, 192, 362
 - software maintenance, 139, 143, 146
 - life cycle framework, 312
- IEEE Std 1220-1998, systems engineering process control, 259, 362

- life cycle processes, 354–356, 354–357
 - compatibility with ISO/IEC 15288, 356–357
 - schematic and outline, 260–262
- IEEE Std 1228-1994:
 - software quality applications, 237–238
 - systems engineering safety, 258–259, 262–263, 362
 - life cycle framework, 351–352
- IEEE Std 1233:
 - system requirements specification, 89, 362
 - life cycle process framework, 301
 - systems engineering design standards, 258
 - requirements specifications, 262, 264
 - system-software integration, 48–49
- IEEE Std 1320.1-1998:
 - IDEF0 standard, 225–226, 362
 - life cycle process, 301–302
 - requirements modeling tools, 220
 - software design notation, 108
 - software engineering process, definition notation, 193
 - software requirements modeling, 85
- IEEE Std 1320.2-1998:
 - IDEFIX 97 development, 226, 362
 - life cycle process, 301–302
 - requirements modeling tools, 220
 - software design notation, 108
 - software requirements modeling, 85
- IEEE Std 1362-1998:
 - Concept of Operations document, 88, 257, 363
 - software acquisition, 297
 - stakeholder requirements, 346–348
 - systems engineering, concept of operations, 264–265
 - system-software integration, 48–49
- IEEE Std 1420.1-1995:
 - asset management, 326
 - Basic Interoperability Data Model, 227, 363
 - software reuse, 222
- IEEE Std 1420.1a-1996:
 - asset certification, 222
 - asset management, 326–327, 363
- IEEE Std 1420.1b-1999, intellectual property rights, 222, 327, 363
- IEEE Std 1462-1998:
 - CASE tool evaluation, 224, 227–228, 363
- infrastructural management
 - framework, 330
 - software quality assessment, 239
- IEEE Std 1465-1998:
 - quality requirements and testing, 319, 363
 - software quality evaluation, 239
 - requirements and testing, 246
 - software requirements specification, 91
- IEEE Std 1471-2000:
 - architectural design, 87
 - life cycle processes, architectural design, 303
 - requirements modeling tools, 220
 - software structure and architecture, 104, 363
 - system design requirements, 258
 - life cycle framework, 350–351
 - systems engineering, architectural requirements, 265–267
- IEEE Std 1490-2003:
 - project management standards, 253
 - life cycle framework, 329
 - software project management, 168
 - standards reconciliation, 178
 - systems engineering PMI standard, 267–269, 364
- IEEE Std 1517-1999:
 - life cycle processes framework, 294, 364
 - acquisition process, 297–298
 - architectural design, 303
 - asset management, 326
 - implementation activity, 300
 - operation process, 311
 - software maintenance, 312
 - system integration, 310
 - system qualification testing, 310
 - software construction process, reuse guidelines, 117–118
 - software engineering process:
 - reuse process, 199, 364
 - reuse standards, 191–192
- IEEE Std 1540-2001:
 - life cycle processes framework, 294
 - risk management, 333–334
 - systems life cycle processes, 344–345
 - software engineering process, risk management, 191–192, 201, 203–204
 - software risk management, 171–172, 364

- IEEE Std 2001-2002, web site management, 277–279, 364
- IEEE Std 14143.1-2000:
 - functional size measurements, 94, 96–97, 365
 - measurement processes, 325
 - software engineering process, process/product measurement, 196
 - software maintenance, 141–142
 - software project management, process monitoring, 174
- Impact analysis:
 - software maintenance, 141
 - software requirements, 94
- Implementation:
 - life cycle process, 300
 - system engineering, 351–352
 - process abstraction, 289
 - processes, 311
 - software engineering process, 185–189
 - Basili model, 190
 - IEEE/EIA 12207.2-1997 standard, 207
 - infrastructure, 186
 - models, 187–188
 - organizational environment, 188
 - plan-do-check-act cycle, 187
 - training issues, 189
- ImproveIT study, software engineering process assessment, 212
- Improvement process, life cycle framework, 330–333
 - CMMI model, 332–333
 - SEI CCM model, 331–332
- INCITS. *See* International Committee for Information Technology Standards
- INCOSE. *See* International Council on Systems Engineering
- INCOSE Systems Engineering Handbook, 33
- Independent testing, software testing process, 132
- Indicators, software engineering process standards, 213–214
- Industrial process measurement and control, system aspects standards, 25
- Industry Technical Agreement (ITA) (IEC), 23–25
- Information management, systems life cycle processes, 345
- Infrastructural management:
 - life cycle framework, 329–330
 - software engineering management and, 167
 - software engineering process, 186
- Initiating, Diagnosing, Establishing, Acting & Learning (IDEAL) model, software engineering process modeling, 188
- Initiation:
 - acquisition process, 296
 - supply process, 298
- Inspection tools, software quality, 223
- Installation testing:
 - life cycle processes framework, 310
 - software testing, 127
- Institute of Electrical and Electronics Engineers (IEEE). *See also* IEEE standards
 - Computer Society Software and Systems Engineering Committee, 38–42
 - ethics of, 232
 - history, 39
 - integration with ISO/IEC JTC 1/SC 7, 41–42
 - organization, 39–41
 - software engineering definitions, 3–5
 - software engineering standards, 14–17
 - standards development and, 34–35
 - Std 610 of, 46–48
- Integrated CASE environments, process tools, 223
- Integrated Product and Process Development (IPPD), improvement process, 332
- Integration, systems engineering life cycle, 352
- Integration requirements:
 - life cycle framework, 308–309
 - software construction, 120
 - system integration, life cycle framework, 310
- Integration testing, software, 127
- Integrity levels:
 - software engineering standards development, 18
 - software quality applications, 237
 - software verification and validation, 236

- system and software products, 272–274
- Interaction issues, software design, 103
- Interdisciplinary approach, software engineering, 5–6
- Interface and implementation
 - separation, software design, 102
- Interface control, software configuration management, 155
- Internal quality:
 - ISO/IEC 9126-1:2001 standard for, 95
 - software products, 233
- Internal testing, software testing process, 132
- International Committee for Information Technology Standards (INCITS), 35–36
- International Council on Systems Engineering (INCOSE), 36
- International Electrotechnical Commission (IEC):
 - IEC SC 65A standard, 25
 - IEC TC 56 Dependability Standard, 24–25
 - IEC TC 62 Electrical Equipment in Medical Practice, 25
 - ISO/IEC JTC 1 standard, 27–28
 - software engineering standards, 10
 - standards development by, 22–25
- International Function Point Users' Group (IFPUG), functional size measurement, 97
- International Organization for Standardization (ISO):
 - ISO/IEC JTC 1 standard, 27–28
 - ISO TC 159, Ergonomics, 26
 - ISO TC 176, Quality management and assurance, 26–27
 - software engineering standards, 10
 - structure and development functions, 25–27
- International Standard (IS) document (IEC), 22–25
- International standards organizations:
 - software engineering standards development, 16–17
 - structure and development functions, 22–28
 - European Committee for Standardization, 28
 - International Electrotechnical Commission, 22–25
 - International Organization for Standardization, 25–27
 - International Telecommunications Union, 28
 - ISO/IEC JTC 1, 27–28
- International Telecommunications Union (ITU):
 - software engineering standards, 10
 - structure and function, 28
- Interpreters, software construction with, 221
- Investment management, systems life cycle, 344
- ISO 9000 standards:
 - collection coordination with S2ESC, 47–48
 - quality management, 254–255
 - software quality management, 233
- ISO 9000:2000, quality management, 254, 364
 - requirements, 271, 364
 - systems engineering, 268, 270
- ISO 9001:2000:
 - application of ISO/IEC 90003:2004, 274–276
 - quality management, 254, 364
 - life cycle processes framework, 319–320
- ISO 9004:2000, quality management, 254, 364
 - performance improvement, 272
- ISO 10007, configuration management, software project process control, 175
- ISO 19011:2002, quality management auditing, 271–272
- ISO 90003:2004:
 - quality management, 255
 - software quality management, 233
- ISO Guide 73, software engineering process risk management, 201
- ISO/IEC 9126 series, software construction measurement, 118
- ISO/IEC 9126-1:2001:
 - software engineering process, 197
 - software maintenance, 145
 - software product quality, 80, 105, 233, 365
 - software requirements standard, 95–96
 - life cycle framework, 305

- ISO/IEC 12207:1995, software engineering process, 204–206
- ISO/IEC 12207:1996:
 - commercial life cycle processes, 286–288
 - documentation standards, 315–316
 - life cycle process, 365
 - process abstraction model, 289–291
 - software engineering process:
 - amendments, 211
 - guides to, 210–211
 - IEEE/EIA 12207 *vs.*, 209–210
 - process/product measurement, 196
 - software structure and architecture, 103–105
- ISO/IEC 14764:1999:
 - software engineering process, 192
 - software maintenance, 143, 146–147, 365
 - life cycle framework, 312
- ISO/IEC 15026:1998:
 - software design, 103
 - software quality applications guidelines, 237
 - software requirements, high-integrity systems, 88
 - system and software integrity levels, 272–274, 366
 - architectural design, life cycle framework, 351
 - systems engineering standards, 259
- ISO/IEC 15288:2002:
 - software engineering process, reference-level process standard, 191–192
 - systems engineering process:
 - control, 259
 - life cycle framework, 274–275, 338, 366
 - agreement process, 344
 - enabling systems, 339
 - enterprise processes, 344–345
 - IEEE Std 1362 correspondence, 346–349
 - models, 340
 - process view, 341–344
 - project processes, 345–346
 - stages, 340, 353
 - system structure, 339
 - technical processes, 346–353
 - system-software integration, 49
 - life cycle framework, 341–344
- ISO/IEC 15504 (draft):
 - improvement process, 331
 - software engineering process, 193
 - assessment, 194–195, 366
 - process/product measurement, 196
- ISO/IEC 15939:2002:
 - life cycle processes framework, 293–294
 - measurement process, 324
 - reference process framework, 291–292
 - software engineering process:
 - measurement standard, 191–192, 213–214, 366
 - process/product measurement, 196
 - software information model, 197
 - software project management, measurement standards, 176–177
 - software quality measurement, 240
- ISO/IEC 90003, S2ESC coordination with, 47–48
- ISO/IEC 90003:2004, application ISO 9001:2000 to software, 274–276, 366
- quality management, 320
- ISO/IEC JTC 1/SC 7 standard:
 - application of ISO/IEC 90003:2004, 275–276
 - collection coordination with S2ESC, 46–49
 - IEEE S2ESC integration with, 41–42
 - IEEE Std 1462-1998 cooperation with, 227–228
 - software and systems engineering, 36–38
 - process framework for, 52–55
 - software engineering process:
 - amendments to ISO/IEC 12207, 211
 - life cycle process standard, 206
 - risk management, 201
 - software project management, 170
 - standards reconciliation, 178
 - software quality management, 233
 - requirements and testing revisions, 246
- ISO/IEC JTC 1/SC 7/WG7 TRs:
 - commercial life cycle processes, 287–288
 - software engineering process standards guide, 210–211
- ISO/IEC JTC 1 standard, development of, 27–28

- ISO/IEC TR 15271, software engineering process:
 - reference-level process standard, 191–192
 - standards guides, 210–211
 - ISO/IEC TR 15504:
 - commercial life cycle processes, 287–288
 - improvement process, 331
 - software engineering process assessment, 211–21
 - ISO/IEC TR 16326, software project management, 170
 - standards reconciliation, 178
 - ISO TC 176:
 - application of ISO/IEC 90003:2004, 275–276
 - collection coordination with S2ESC, 47–48
 - Quality management and assurance, 26–27
 - quality management improvement, 272
 - Issue tools, software configuration with, 222
 - ITU. *See* International Telecommunications Union
- J**
- Joint review process, life cycle framework, 322–323
 - Joint technical committees (JTCs), ISO/IEC JTC 1 standard, 27–28
- K**
- Knowledge areas:
 - computer engineering, 250–251
 - computer science, 251
 - management, 252
 - mathematics, 252
 - project management, 252–253
 - quality management, 253–254
 - software configuration management:
 - auditing, 160–163
 - constraints and guidance, 153
 - control issues, 158–159
 - identification, 156–158
 - organizational context, 152–153
 - planning protocols, 153–155
 - process management, 152
 - standards, 163
 - status accounting, 159–160
 - surveillance, 155
 - SWEBOK scope and summary, 149–150
 - software construction, 113–121
 - basic concepts, 115–116
 - coding requirements, 119
 - design issues, 118
 - integration guidelines, 120
 - language requirements, 119
 - management guidelines, 116–118
 - quality controls, 119
 - standards, 120–121
 - SWEBOK scope and summary, 114–115
 - testing protocols, 119
 - software design, 99–111
 - fundamentals, 101–102
 - key issues, 102–103
 - notations, 107–108
 - quality analysis and evaluation, 105–107
 - scope, 99–100
 - standards, 109–111
 - strategies and methods, 108–109
 - structure and architecture, 103–105
 - summary, 100–101
 - software engineering management:
 - closure, 176
 - initiation and scope definition, 168–169
 - measurement, 176–177
 - planning, 169–172
 - deliverables, 170–171
 - effort, schedule, and cost estimation, 171
 - plan management, 172
 - process planning, 170
 - quality management, 172
 - resource estimation, 171
 - risk management, 171–172
 - project enactment, 173–175
 - process monitoring, 173–175
 - supplier contract management, 173
 - review and evaluation, 175
 - standards, 177–180
 - SWEBOK scope and summary, 165–167
 - software engineering methods:
 - formal methods, 224
 - heuristic methods, 224
 - prototyping, 224
 - standards, 225–228

Knowledge areas (*continued*)

- software engineering process:
 - adaptation, 194
 - assessment, 194–195
 - automation, 194
 - conceptual-level standards, 192–193
 - definition, 189–194
 - notation, 193
 - implementation and change,
 - 185–189
 - models for, 187–188
 - organizational environment, 188
 - plan-do-check-act cycle, 187
 - process infrastructure, 186
 - training, 189
 - life cycle models, 190
 - life cycle processes, 190–193
 - measurement, 195–198
 - information models, 197
 - process measurement, 196
 - product measurement, 196–197
 - quality of results, 197
 - techniques, 197–198
 - reference-level standards, 191–192
 - standards for, 198–214
 - IEEE/EIA 12207.1-1996, 206–207
 - IEEE/EIA 12207.2-1997, 207
 - IEEE/EIA 12207-1996, 201–202, 204–209
 - IEEE Std 1074-1997, 198–201
 - IEEE Std 1517-1999, 199
 - IEEE Std 1540-2001, 201, 203–204
 - ISO/IEC 12207 TR guides, 210–211
 - ISO/IEC 12207 *vs.* IEEE/EIA 12207, 209–210
 - ISO/IEC 15939:2002, 213–214
 - ISO/IEC amendments, 211
 - ISO/IEC draft 15504, 212–213
 - ISO/IEC TR 15504, 211–212
 - SWEBOK scope and summary, 183–185
- software engineering tools:
 - configuration management tools, 221–222
 - construction tools, 220–221
 - design tools, 220
 - evaluation of, 223–224
 - maintenance tools, 221
 - management tools, 222
 - meta-tools, 223
 - process tools, 222–223
 - quality tools, 223
 - requirements tools, 219–220
 - standards, 225–228
 - SWEBOK scope and summary, 217–218
 - testing tools, 221
 - tool integration techniques, 223
- software maintenance, 137–148
 - activities, 143–145
 - cost estimation, 141
 - fundamental categories, 139–140
 - management issues, 141
 - measurement programs, 142
 - process models, 142–143
 - standards, 146–148
 - supporting activities, 144–145
 - SWEBOK scope and summary, 137–138
 - technical issues, 141
 - techniques for, 145–146
 - unique activities, 143–144
- software quality:
 - application requirements, 237–238
 - defect characterization, 238
 - engineering culture and ethics, 231–232
 - fundamental principles, 231
 - improvement programs, 234
 - management processes, 234
 - management techniques, 238–239
 - measurement, 239–240
 - model characteristics, 232–234
 - product quality, 233
 - quality assurance, 235
 - reviews and audits, 236–237
 - standards, 240–246
 - SWEBOK scope and summary, 230
 - value and cost issues, 232
 - verification and validation, 235–236
- software requirements, 77–98
 - analysis, 84–88
 - architectural design, 85–87
 - classification, 84–85
 - conceptual modeling, 85
 - high-integrity systems, 88
 - negotiation, 87
 - elicitation, 84
 - fundamental principles, 79–82
 - practical considerations, 93–94
 - requirements process, 82–84
 - scope, 77

- specification, 88–92
 - software requirements description, 90–92
 - system definition document, 88
 - system requirements, 89
 - standards, 94–98
 - IEEE Std 830-1998, 94–95
 - IEEE Std 14143.1-2000, 96–97
 - ISO/IEC 9126-1:2001, 95–96
 - summary, 78–79
 - validation requirements, 92–93
 - software testing, 123–136
 - applications-based, 130
 - basic concepts and definitions, 125–127
 - code-based, 129
 - deterministic *vs.* random, 130
 - experience-based testing, 128
 - fault-based, 129
 - functional *vs.* structural, 130
 - levels of testing, 127
 - measurement tools, 130–131
 - objectives of, 127–128
 - process management, 132–134
 - specification-based, 128–129
 - standards sources, 134–136
 - SWEBOK scope and summary, 123–125
 - techniques for, 128–130
 - usage-based, 129–130
 - standards organization based on, 67–69
 - systems engineering, 256–257
 - Knowledge-based organization, of
 - SWEBOK guide, 74–76
- L**
- Language requirements, software construction, 119
 - Levels of testing, software testing, 127–128
 - Liability issues, software engineering standards, 14
 - Life cycle data, software engineering process, standards for, 206–207
 - Life cycle modeling, software engineering process, 190
 - Life cycle processes standards:
 - commercial standards, 286–288
 - cross-project processes, 334–335
 - domain (reuse) engineering, 334–335
 - DoD standards, 283–285
 - harmonization of, 286–288
 - history and concepts, 283–288
 - IEEE/EIA 12207 standard:
 - framework of, 293–295
 - software design process, 102
 - organizational processes, 327–334
 - improvement, 330–333
 - infrastructure, 329–330
 - management, 327–329
 - reuse program administration, 334
 - risk management, 333–334
 - training, 333
 - organization of, 64
 - primary processes, 294–298
 - acquisition, 296–298
 - development process, 299–313
 - acceptance support, 311
 - analysis activity, 300–302
 - architectural design activity, 302–303
 - coding and testing activity, 307–308
 - implementation activity, 300
 - installation, 310
 - integration, 308–309
 - qualification testing, 309–310
 - software architecture design, 305–306
 - software detailed design, 306–307
 - software requirements analysis, 303–305
 - system integration, 310
 - system qualification testing, 310
 - maintenance, 311–313
 - operation process, 311
 - supply process, 298–299
 - software construction process, reuse guidelines, 117–118
 - software engineering process, 190–193, 201–206
 - conceptual level process standards, 192–193
 - reference-level process standards, 191–192
 - software maintenance, 139–144
 - supporting processes, 313–327
 - asset (reuse) management, 325–327
 - auditing, 323
 - configuration management, 317–318
 - documentation, 314–316
 - joint review, 322–323
 - measurement, 324–325

- Life cycle processes standards (*continued*)
 - problem resolution, 323–324
 - quality assurance, 318–320
 - validation, 321–322
 - verification, 320–321
- systems engineering, 274–275
 - agreement processes, 344
 - cycle stages, 340
 - enabling systems, 339
 - enterprise processes, 344–345
 - IEEE 1220 and, 354–357
 - ISO/IEC 15288-IEEE 1220
 - compatibility, 356–357
 - model, 340
 - overview, 337–338
 - processes, 340–341
 - project processes, 345–346
 - SEP protocols, 355–356
 - software correspondence, 341–344
 - staged perspective, 353
 - structure, 339
 - technical processes, 346–353
 - architectural design, 350–351
 - disposal, 353
 - implementation, 351–352
 - integration, 352
 - maintenance, 353
 - operation, 353
 - requirements analysis, 347–350
 - stakeholder requirements, 346–347
 - transition, 353
 - validation, 353
 - verification, 352
- Life test, software testing, 131
- Limited understanding, software maintenance, 141
- M**
- Maintainability:
 - in product quality model, 60
 - software maintenance, 141
- Maintenance process, life cycle processes framework, 311–313
 - systems engineering, 353
- Malfunction analysis, software testing, 126
- Management guidelines:
 - approval, for software engineering standards, 10–11
 - defined, 165
 - knowledge areas, 252
 - life cycle framework, 327–329
 - software maintenance, 141
 - software quality management process, 234
 - software testing process, 132–134
- Mathematics, knowledge areas, 252
- Measurable concept, software engineering process standards, 213–214
- Measurement information model, software engineering process, 213
- Measurement standards:
 - organization of, 65
 - software construction, 118
 - software design quality, 107
 - software engineering criteria, 13
- Measurement techniques:
 - life cycle framework, 324–325
 - software engineering process measurement, 195–198
 - software maintenance, 142
 - software project management, 176–177
 - software quality, 239–240
 - software testing, 130–132
- Medical devices, software and software life cycle processes, standards for, 25
- Medical practice, technical equipment standard for (IEC), 25
- Meta-tools, defined, 223
- Migration, maintenance process and, 312
- Mil-Std 498 standard, 285
- Modeling tools:
 - software quality, 232
 - software requirements, 219–220
 - systems engineering, life cycle framework, 340
- Model validation, software requirements, 93
- Modification analysis, life cycle processes framework, 311
- Modularity principle, systems engineering, life cycle framework, 340–341
- Multiple supplier programs, software engineering process standards conformance, 208
- Mutation scoring, software testing evaluation, 131
- Mutation testing, fault-based software testing, 129

N

- National Institute of Standards and Technology (NIST):
 - IDEF modeling language standard, 225–226
 - software engineering standards development, 16–17
 - Needs definition standards, system engineering, 257
 - Negotiation, software requirements, 87
 - Nomenclature, software engineering standards, 12–13
 - Nonfunctional requirements:
 - architectural design and, 87
 - defined, 80
 - Notation:
 - software design, 107–108
 - software engineering process definition, 193
- O**
- Object-based standards organization, 66–67
 - Occupational definitions, software engineering, 4–5
 - Openness, on software engineering standards, 10
 - Operational environment:
 - life cycle processes framework, 311
 - systems engineering, 353
 - software requirement elicitation, 84
 - Operational profile, usage-based software testing, 130
 - Operational testing, life cycle processes, 311
 - Organizational environment:
 - software configuration management, 152–153
 - responsibilities, 154
 - software engineering management, 167
 - software engineering process:
 - conformance to standards, 208
 - infrastructure management, 186
 - software engineering standards, 52
 - software maintenance, 141
 - software quality, engineering culture and ethics, 231–232
 - software requirements elicitation, 84
 - Organizational goals, software engineering standards development, 17–18

- Organizational life cycle processes, 294–295, 327–334
 - improvement, 330–333
 - infrastructure, 329–330
 - management, 327–329
 - reuse program administration, 334
 - risk management, 333–334
 - training, 333
 - Organizationally adopted processes, S2ESC emphasis on, 49–51
 - Ownership principle, systems engineering, life cycle framework, 340–341
- P**
- People-intensive analysis, software quality management, 238
 - Perfective software maintenance, 140
 - Performance analysis tools, software testing, 221
 - Performance evaluation, software testing, 128
 - Performance improvement standards, quality management, 272
 - Personal software process, measurement, 198
 - Physical Configuration Audit (PCA), software configuration management, 161–162
 - “Pink Ballot” (EIA standard). *See* Electronic Industries Alliance
 - Plan-Do-Check-Act (PDCA) cycle:
 - life cycle framework, management role, 327–329
 - software engineering process, 187
 - Planning aids:
 - management tools, software project management, 172
 - software configuration management, 153, 155
 - software maintenance, 144–145
 - software testing, 130
 - Plans’ standards, organization of, 65
 - PMBOK Guide*, 170
 - PMI. *See* Project Management Institute
 - Portability, in product quality model, 60
 - Post-implementation support, systems engineering, 259
 - Predictive validity, software engineering process measurement, 197
 - Prescription, levels of, standards organization based on, 66–67

- Presentation issues, software design, 103
- Preventive software maintenance, 140
- Primary life cycle processes, 295–313
 - acquisition, 296–298
 - development process, 299–313
 - acceptance support, 311
 - analysis activity, 300–302
 - architectural design activity, 302–303
 - coding and testing activity, 307–308
 - implementation activity, 300
 - installation, 310
 - integration, 308–309
 - qualification testing, 309–310
 - software architecture design, 305–306
 - software detailed design, 306–307
 - software requirements analysis, 303–305
 - system integration, 310
 - system qualification testing, 310
 - maintenance, 311–313
 - operation process, 311
 - supply process, 298–299
- Primary process framework, software engineering standards, 52
- Primitiveness, software design, 102
- Principles and practices, in software engineering, 6–8
- Principle standards, organization of, 67
- Problem reporting:
 - life cycle processes framework, 311
 - software testing, 133
- Problem resolution:
 - life cycle framework for, 323–324
 - software maintenance, 144
- Procedure, process *vs.*, 289–290
- Process abstraction model:
 - basic components, 288–291
 - software engineering standards, 61–62
- Process-based standards organization, 69–70
- Process-centered software engineering environments, tools for, 223
- Process control:
 - software project management, 174–175
 - systems engineering, 259
- Process definition, software engineering process, 189–190
 - review, 198
- Process implementation, life cycle processes framework, 311
- Process models:
 - software maintenance, 142–143
 - tools for, 223
- Process monitoring:
 - software maintenance, 141
 - software project management, 173–174
 - software quality management, 232–233
 - software testing process, 132
 - tools for, 223–224
 - uniform process framework, 52–55
- Process planning, software project management, 170
- Process requirements:
 - defined, 80
 - for software design, 102
 - software engineering, 82–84
 - history and concepts, 283–288
 - reference framework, 291–292
- Product control, systems engineering, 259
- Product evaluation:
 - quality management, 233
 - software engineering process measurement, 195–198
 - software engineering standards development, 18
- Product improvement, software engineering standards, 12–13
- Productivity metrics, software project management, 177
- Product lines, software design, 105
- Product quality model:
 - architectural coherence in software engineering and, 59–60
 - ISO/IEC 9126-1:2001 standard, 80
- Product realization, system engineering, 258
- Product requirements, defined, 80
- Professional discipline, software engineering standards, 14
- Program comprehension, software maintenance, 146
- Program editors, software construction with, 220
- Program families, software design, 105
- Program Management Institute (PMI), systems engineering, 267–268
- Programming languages, software construction, 119

- Project Initiation Notification System (PINS) (ANSI), 30
 - Project-level conformance, software engineering process standards, 208
 - Project management:
 - defined, 165
 - knowledge areas, 252–253
 - systems engineering, PMI standard, 267–268
 - standards for, 253
 - Project Management Institute (PMI):
 - life cycle framework, 329
 - S2ESC coordination with, 47–48
 - structure and function, 36
 - Project management standards, organization of, 65
 - Project planning, software project management, 169–171
 - Project processes, systems life cycle management, 344
 - Prototyping:
 - engineering methods for, 224
 - validation, 93
 - Publicly Available Specification (PAS) (IEC), 23–25
- Q**
- Qualification testing, life cycle framework, 309–310
 - system qualification testing, 310
 - Quality Improvement Paradigm (QIP), software engineering process modeling, 187–188
 - Quality improvement programs, software quality, 234
 - Quality in use, software products, 233
 - Quality management and assurance:
 - ISO/IEC 9126-1:2001 standard for, 96
 - ISO TC 176 standard, 26
 - knowledge areas, 253–254
 - life cycle processes, 318–320
 - product quality model, architectural coherence in software engineering and, 59–60
 - scope, 253
 - software construction, 119
 - software design, 105–106
 - software engineering process, measurement results, 197
 - software engineering projects, 172
 - software maintenance, 145
 - software quality:
 - application requirements, 237–238
 - defect characterization, 238
 - engineering culture and ethics, 231–232
 - fundamental principles, 231
 - improvement programs, 234
 - management processes, 234
 - management techniques, 238–239
 - measurement, 239–240
 - model characteristics, 232–234
 - product quality, 233
 - quality assurance, 235
 - reviews and audits, 236–237
 - standards, 240–246
 - SWEBOK guidelines on, 75
 - SWEBOK scope and summary, 230
 - tools for, 223
 - value and cost issues, 232
 - verification and validation, 235–236
 - software requirements, 80
 - standards, 254–255
 - SWEBOK guidelines on, 75
 - systems life cycle management, 344
 - tools for, 223
- Quality model, software engineering standards development, 20
- R**
- Random testing:
 - software evaluation, 130
 - specification-based software testing, 129
 - Recommended Practices (IEEE), software structure and architecture, 104–105
 - Recovery testing, software testing, 128
 - Re-engineering:
 - software maintenance, 146
 - tools for, 221
 - Reference-level process standards, software engineering processes, 191–192
 - Reference processes:
 - framework for, 291–292
 - process abstraction, 288
 - software engineering and, 190
 - Regulatory conformance, software engineering process standards, 208

Release and build tools, software configuration with, 222

Reliability:

- in product quality model, 60
- software engineering process measurement, 197
- software testing, 128
 - growth models, 131
 - life tests, 131

Request for proposal (RFP), acquisition process, 296

Requirements:

- defined, 80
- engineering, software requirements fundamentals, 79–82
- management practices, 93–94
- modeling tools, 220
- specification, defined, 88
- system life cycle analysis, 347–350
- tracing, software requirements, 82

Resources:

- software configuration management, guidelines for, 154
- software project management, estimation of, 171
- systems life cycle management, 344

Responsibilities, software configuration management, guidelines for, 154

Reuse standards:

- asset management, 325–327
- cross-project processes, 334–335
- organization of, 65
- program administration, 334
- software testing, 132

Reverse engineering, software maintenance, 146

Reviews:

- software project management, 175
- software quality, 236–237
- software requirements, 93

Risk management:

- life cycle framework for, 333–334
- software project management and, 171–172
- systems life cycle processes, 344–345

Robustness testing, specification-based software testing, 129

Root cause analysis, software engineering process measurement, 198

RTCA DO-178B, commercial life cycle standard, 286–288

S

Safety issues:

- life cycle framework, systems engineering, 351–352
- software quality applications, 237–238

Scheduling estimation, software project management, 171

Scheduling guidelines, software configuration management, 154

Scope:

- for software engineering standards, 11–12
- software requirements knowledge area, 77

SDOs. *See* Standards developing organizations

Software and Systems Engineering Standards Committee (S2ESC) (IEEE):

- collection principles, 45–62
 - architectural coherence, 58–61
 - consistency issues, 59
 - data framework, 55–58
 - organizationally adopted processes, 49–51
 - process abstraction model, 61–62
 - product quality model, 59–60
 - software-systems integration, 48–49
 - strategic relationships, 45–48
 - uniform process framework, 51–55
- history, 39
- integration with ISO/IEC/JTC 1/SC 7, 41–42
- knowledge areas organization, 67–69
- levels of prescription, 67–68
- objects of, 67
- organization, 39–41
- structure and activities of, 35

Software Capability Evaluation (SCE)

- methods, software engineering process assessment, 194–195

Software Capability Maturity Model (SW-CMM):

- improvement process, 331–332
- software engineering process, 194–195

Software coding, life cycle framework, 307–308

Software configuration:

- SWEBOK guidelines on, 75
- tools for, 221–222

Software configuration management (SCM):

- auditing, 160–163
- constraints and guidance, 153
- control issues, 158–159
- identification, 156–158
- organizational context, 152–153
- planning protocols, 153–155
- process management, 152
- standards, 163
- status accounting, 159–160
- surveillance, 155
- SWEBOK scope and summary, 149–150
- Software construction, 113–121
 - basic concepts, 115–116
 - coding requirements, 119
 - design issues, 118
 - integration guidelines, 120
 - language requirements, 119
 - management guidelines, 116–118
 - quality controls, 119
 - standards, 120–121
 - SWEBOK guidelines on, 75
 - SWEBOK scope and summary, 114–115
 - testing protocols, 119
 - tools for, 220–221
- Software design:
 - knowledge area, 99–111
 - fundamentals, 101–102
 - key issues, 102–103
 - notations, 107–108
 - quality analysis and evaluation, 105–107
 - scope, 99–100
 - standards, 109–111
 - strategies and methods, 108–109
 - structure and architecture, 103–105
 - summary, 100–101
 - SWEBOK guidelines on, 74
 - tools for, 220
- Software engineering:
 - basic principles, 6–8
 - culture and ethics in, 231–232
 - defined, 3–5
 - interdisciplinary approach, 5
 - knowledge areas, 5–6, 256–257
 - management:
 - closure, 176
 - initiation and scope definition, 168–169
 - measurement, 176–177
 - planning, 169–172
 - deliverables, 170–171
 - effort, schedule, and cost estimation, 171
 - plan management, 172
 - process planning, 170
 - quality management, 172
 - resource estimation, 171
 - risk management, 171–172
 - project enactment, 173–175
 - process monitoring, 173–175
 - supplier contract management, 173
 - review and evaluation, 175
 - standards, 177–180
 - SWEBOK guidelines on, 75
 - SWEBOK scope and summary, 165–167
 - tools, 222
- methods:
 - formal methods, 224
 - heuristic methods, 224
 - prototyping, 224
 - standards, 225–228
- process:
 - abstraction model, 288–291
 - adaptation, 194
 - assessment, 194–195
 - automation, 194
 - conceptual-level standards, 192–193
 - definition, 189–194
 - notation, 193
 - history and concepts, 283–288
 - implementation and change, 185–189
 - models for, 187–188
 - organizational environment, 188
 - plan-do-check-act cycle, 187
 - process infrastructure, 186
 - training, 189
 - life cycle models, 190
 - life cycle processes, 190–193
 - measurement, 195–198
 - information models, 197
 - process measurement, 196
 - product measurement, 196–197
 - quality of results, 197
 - techniques, 197–198
 - quality management, 232–233
 - reference framework, 291–292
 - reference-level standards, 191–192
 - standards for, 198–214
 - IEEE/EIA 12207.1-1996, 206–207

Software engineering (*continued*)

- IEEE/EIA 12207.2-1997, 207
- IEEE/EIA 12207-1996, 201–202, 204–209
- IEEE Std 1074-1997, 198–201
- IEEE Std 1517-1999, 199
- IEEE Std 1540-2001, 201, 203–204
- ISO/IEC 12207 TR guides, 210–211
- ISO/IEC 12207 *vs.* IEEE/EIA 12207, 209–210
- ISO/IEC 15939:2002, 213–214
- ISO/IEC amendments, 211
- ISO/IEC draft 15504, 212–213
- ISO/IEC TR 15504, 211–212
- SWEBOK guidelines on, 75
- SWEBOK scope and summary, 183–185
- tools, 222–223
- standards, 8–19, 36–42
 - business protection, 14
 - buyer protection, 13
 - collection integration, ISO/IEC JTC 1/SC 7-IEEE S2ESC, 41–42
 - development of, 15–17
 - history of, 15
 - IEEE Computer Society Software and Systems Engineering Committee, 38–41
 - ISO/IEC/JTC 1/SC 7, 36–38
 - nature of, 9–11
 - organizational goals for, 17–18
 - product improvement, 12–13
 - professional discipline, 14
 - scope of, 11–12
 - technology development, 14
 - trends in, 18–19
- SWEBOK scope concerning, 73–74, 256
- terminology, 277–278
- tools:
 - configuration management tools, 221–222
 - construction tools, 220–221
 - design tools, 220
 - evaluation of, 223–224
 - maintenance tools, 221
 - management tools, 222
 - meta-tools, 223
 - process tools, 222–223
 - quality tools, 223
 - requirements tools, 219–220
 - standards, 225–228
 - SWEBOK guidelines on, 75
 - SWEBOK scope and summary, 217–218
 - testing tools, 221
 - tool integration techniques, 223
- Software Engineering Institute (SEI), Capability Maturity Model, 331–332
- Software Engineering Process Group (SEPG), software engineering process infrastructure, 186
- Software Engineering Standards Committee (SESC), history of, 15
- Software ergonomics, 255–256
- Software information models, software engineering process measurement, 197
- Software integration:
 - life cycle framework, 308–309
 - software construction process, 116–117
- Software Life Cycle Processes*, 14
- Software maintenance:
 - knowledge areas, 137–148
 - activities, 143–145
 - cost estimation, 141
 - fundamental categories, 139–140
 - management issues, 141
 - measurement programs, 142
 - process models, 142–143
 - standards, 146–148
 - supporting activities, 144–145
 - SWEBOK scope and summary, 137–138
 - technical issues, 141
 - techniques for, 145–146
 - unique activities, 143–144
- SWEBOK guidelines on, 75
- tools for, 221
- Software Process Improvement and Capability dEtermination (SPICE), software engineering process assessment, 212
- Software project management, basic principles, 168
- Software project management plan (SPMP), standard for, 178
- Software qualification testing, life cycle framework, 309–310

- Software quality:
 - application requirements, 237–238
 - defect characterization, 238
 - engineering culture and ethics, 231–232
 - fundamental principles, 231
 - improvement programs, 234
 - management processes, 234
 - management techniques, 238–239
 - measurement, 239–240
 - model characteristics, 232–234
 - product quality, 233
 - quality assurance, 235
 - reviews and audits, 236–237
 - standards, 240–246
 - SWEBOK guidelines on, 75
 - SWEBOK scope and summary, 230
 - tools for, 223
 - value and cost issues, 232
 - verification and validation, 235–236
- Software Quality Assurance Plan (SQAP), requirements for, 241–242
- Software reliability-engineered testing (SRET), usage-based software testing, 130
- Software requirements:
 - knowledge area, 77–98
 - analysis, 84–88
 - architectural design, 85–87
 - classification, 84–85
 - conceptual modeling, 85
 - high-integrity systems, 88
 - negotiation, 87
 - elicitation, 84
 - fundamental principles, 79–82
 - organization and schematic for, 79
 - practical considerations, 93–94
 - requirements process, 82–84
 - specification, 88–92, 90–91
 - standards, 94–98
 - SWEBOK scope and summary, 77–79
 - life cycle processes framework, 303–305
 - specification:
 - software requirements description, 90–92
 - system definition document, 88
 - system requirements, 89
 - standards:
 - IEEE Std 830-1998, 94–95
 - IEEE Std 14143.1-2000, 96–97
 - ISO/IEC 9126-1:2001, 95–96
 - SWEBOK guidelines on, 74
 - tools for, 219–220
 - validation requirements, 92–93
- Software Requirements Description, 82
- Software retirement, maintenance process and, 312
- Software Safety Plans*, 14
- Software testing:
 - knowledge areas, 123–136
 - applications-based, 130
 - basic concepts and definitions, 125–127
 - code-based, 129
 - deterministic *vs.* random, 130
 - experience-based testing, 128
 - fault-based, 129
 - functional *vs.* structural, 130
 - levels of testing, 127
 - measurement tools, 130–131
 - objectives of, 127–128
 - process management, 132–134
 - specification-based, 128–129
 - standards sources, 134–136
 - SWEBOK scope and summary, 123–125
 - techniques for, 128–130
 - usage-based, 129–130
 - life cycle framework, 307–308
 - SWEBOK guidelines on, 75
 - tools for, 221
- Specifications:
 - data framework for software standards, 56–58
 - in software requirements, 88
 - software testing, 128–129
 - system requirements, 89
- Stability measures, software maintenance, 145
- Staffing issues, software maintenance, 141
- Staged capability model, software engineering process, 194–195
- Stages perspective, systems engineering, life cycle framework, 340
- Stakeholder requirements, systems engineering life cycle, 346–347
- Standard Quality Assurance Plan*, history of, 15

- Standards developing organizations (SDOs):
 - international standards organizations, 22–28
 - European Committee for Standardization, 28
 - International Electrotechnical Commission, 22–25
 - International Organization for Standardization, 25–27
 - International Telecommunications Union, 28
 - ISO/IEC JTC 1, 27–28
 - overview of, 21–36
 - software and systems engineering standards, 36–42
 - collection integration, ISO/IEC JTC 1/SC 7-IEEE S2ESC, 41–42
 - IEEE Computer Society Software and Systems Engineering Committee, 38–41
 - ISO/IEC/JTC 1/SC 7, 36–38
 - software engineering standards, 8–9
 - United States organizations, 28–36
 - American Institute of Aeronautics and Astronautics, 30–32
 - American National Standards Institute, 29–30
 - Electronic Industries Alliance, 32–34
 - Institute of Electrical and Electronics Engineers, 34–35
 - InterNational Committee for Information Technology Standards, 35–36
 - International Council on Systems Engineering, 36
 - Project Management Institute, 36
- Standards organization:
 - by knowledge area, 67–69
 - object-based organization, 66–67
 - process-based organization, 69–70
 - software maintenance, 146–147
 - topic-based organization, 63–66
 - documentation standards, 64
 - life cycle processes standards, 64
 - measurement standards, 65
 - plans standards, 65
 - project management standards, 65
 - reuse standards, 65
 - terminology standards, 65
 - tools standards, 66
- Standards sources:
 - software construction knowledge area, 129–131
 - software engineering, 8–19, 36–42
 - business protection, 14
 - buyer protection, 13
 - collection integration, 41–42
 - development of, 15–17
 - history of, 15
 - IEEE Computer Society Software and Systems Engineering Committee, 38–41
 - ISO/IEC/JTC 1/SC 7, 36–38
 - nature of, 9–11
 - organizational goals for, 17–18
 - product improvement, 12–13
 - professional discipline, 14
 - scope of, 11–12
 - technology development, 14
 - terminology, 277
 - trends in, 18–19
 - web site engineering, 277–279
 - Static analysis tools, software quality, 223, 238
 - Statistical process control, software engineering process measurement, 198
 - Strategies and methods, software design, 108–109
 - Structural analysis, systems engineering, life cycle framework, 339
 - Structural testing, software evaluation, 130
 - Structural theory, software design notation, 107
 - Structure, software design and, 103–105
 - Subcommittee (SC) of IEC:
 - IEC SC 65A standard, 25
 - structure and function, 23–24
 - Subcontractor control, software configuration management, 155
 - Sufficiency, software design, 102
 - Supplier contract management:
 - acquisition process, 296
 - software engineering process standards conformance, 208
 - software projects, 173
 - supply process, 298
 - Supply process:
 - framework for, 298–299
 - life cycle framework, systems agreement, 344

- Supporting activities, software maintenance, 144–145
- Supporting processes, life cycle processes standards, 313–327
 - asset (reuse) management, 325–327
 - auditing, 323
 - configuration management, 317–318
 - documentation, 314–316
 - joint review, 322–323
 - measurement, 324–325
 - problem resolution, 323–324
 - quality assurance, 318–320
 - validation, 321–322
 - verification, 320–321
- Supporting process framework, software engineering standards, 52
- Surveillance guidelines, software configuration management, 155
- SWEBOK. *See Guide to the Software Engineering Body of Knowledge*
- System definition, document for, 88
- System integration, life cycle framework, 310
- System-of-interest, life cycle framework, 339
- System operation, life cycle processes, 311
- System qualification, life cycle framework, 310
- System requirements:
 - life cycle process framework, 300–302
 - software requirements and, 80–81
 - specification, 89
- Systems engineering:
 - knowledge areas, 256–257
 - life cycle processes, 274–275
 - agreement processes, 344
 - cycle stages, 340
 - enabling systems, 339
 - enterprise processes, 344–345
 - IEEE 1220 and, 354–357
 - ISO/IEC 15288-IEEE 1220
 - compatibility, 356
 - model, 340
 - overview, 337–338
 - processes, 340–341
 - project processes, 345–346
 - SEP protocols, 355–356
 - software correspondence, 341–344
 - staged perspective, 353
 - structure, 339
 - technical processes, 346–353
 - architectural design, 350–351
 - disposal, 353
 - implementation, 351–352
 - integration, 352
 - maintenance, 353
 - operation, 353
 - requirements analysis, 347–350
 - stakeholder requirements, 346–347
 - transition, 353
 - validation, 353
 - verification, 352
 - process-based organization, 69–70
 - standards, 257–276
 - design, 257–258
 - needs definition, 257
 - post-implementation support, 259
 - process control, 259
 - product control, 259
 - product realization, 258
 - technical analysis and evaluation, 258–259
 - technical management, 257
 - SWEBOK scope on, 256
 - Systems Engineering Process (SEP):
 - defined, 260
 - IEEE Std 1220-1998 and, 355–336
 - life cycle framework, 356
 - System-software integration, S2ESC strategies for, 48–49
 - Systems-software integration, life cycle framework, 341–344
 - System stakeholders, software requirement elicitation from, 84

T

 - Tailoring of standards:
 - approaches to, 51
 - software engineering process:
 - conformance, 208
 - ISO/IEC 12207 vs. IEEE/EIA 12207-1996, 210
 - Technical analysis and evaluation, system engineering, 258–259
 - Technical approval, for software engineering standards, 10–11
 - Technical Committees (TCs) (IEC):
 - IEC 72304, Medical device software-Software life cycle processes, 25
 - IEC TC 56 Dependability Standard, 24–25
 - structure and function, 23–25
 - IEC TC 62 Electrical Equipment in Medical Practice, 25

- Technical Committees (TCs) (ISO), structure and development functions, 26–27
 - Technical life cycle processes, systems engineering, 346–353
 - architectural design, 350–351
 - disposal, 353
 - implementation, 351–352
 - integration, 352
 - maintenance, 353
 - operation, 353
 - requirements analysis, 347–350
 - stakeholder requirements, 346–347
 - transition, 353
 - validation, 353
 - verification, 352
 - Technical Report (TR) (IEC), 23–25
 - Technical specification (TS) document (IEC), 23–25
 - Technique standards, organization of, 67
 - Technology transition, software engineering standards, 14
 - Technology Trend Assessment (TTA) (IEC), 23–25
 - Termination criteria, software testing, 132
 - Terminology standards:
 - IEEE Std 610.12, 69
 - IEEE Std 2001 *Internet Best Practices*, 69
 - levels of prescription, 66
 - organization of, 65
 - software engineering, 277–278
 - Testability measures, software maintenance, 145
 - Test-driven software development, 128
 - Test evaluation tools, software testing, 221
 - Test execution frameworks, software testing, 221
 - Test generators, software testing, 221
 - Testing. *See also* specific types of testing, e.g. Software qualification testing
 - software construction, 119
 - software maintenance and, 141
 - tools for, 221
 - Test management tools, software testing, 221
 - Test objectives, software testing, 126–128
 - Test targets, software testing, 127
 - Tool evaluation, 223–224
 - Tool integration techniques, 223
 - Toolkit languages, software construction, 119
 - Tool selection and implementation, software configuration management, guidelines for, 154–155
 - Tools standards:
 - contents of, 225–228
 - organization of, 66
 - software construction, 120
 - Top-down approach, software testing, 127
 - Topical access, in SWEBOK guide, 74
 - Topic-based standards organization, 63–66
 - documentation standards, 64
 - life cycle processes standards, 64
 - measurement standards, 65
 - plans standards, 65
 - project management standards, 65
 - reuse standards, 65
 - terminology standards, 65
 - tools standards, 66
 - Traceability:
 - requirements tools for, 219–220
 - software requirements, 94
 - Tracking tools, software configuration with, 222
 - Training:
 - life cycle framework for, 333
 - software engineering process, 189
 - Transition process, life cycle framework, systems engineering, 353
 - Trends analysis, software engineering standards development, 18–19
 - Two-party agreements, software engineering standards development, 17–18
- ## U
- Unified Modeling Language (UML), software design notation, 108
 - Uniform process framework, software engineering standards, 51–58
 - Unique activities, software maintenance, 143–144
 - United States, software engineering standards development, 16–17
 - United States standards organizations, structure and function, 28–36

- American Institute of Aeronautics and Astronautics, 30–32
 - American National Standards Institute, 29–30
 - Electronic Industries Alliance, 32–34
 - Institute of Electrical and Electronics Engineers, 34–35
 - InterNational Committee for Information Technology Standards, 35–36
 - International Council on Systems Engineering, 36
 - Project Management Institute, 36
 - Universality of management principle, 165
 - systems engineering, PMI standard, 267–268
 - Usability:
 - in product quality model, 60
 - software testing, 128–130
 - User requirements:
 - life cycle framework, 306
 - software requirements and, 80–81
 - User support, life cycle processes, 311
- V**
- Validation:
 - life cycle process framework, 321–322
 - systems engineering, 353
 - software construction for, 115
 - software quality, 235–236
 - standards for, 242–244
 - software requirements, 92–93
 - Validity, software engineering process measurement, 197
 - Value, of software quality, 232
 - Vendor control, software configuration management, 155
 - Verifiability, software requirements, 81
 - Verification process:
 - life cycle framework, 320–321
 - systems engineering, 352
 - software quality, 235–236
 - standards for, 242–244
 - software requirements, 92
 - Version management tools, software configuration with, 222
- W**
- Weapons System Software Development*, 285
 - Web site management, standards for, 277–279
 - Working groups (WG) (IEC):
 - function of, 23–25
 - software engineering standards, 37–38
 - Work products, software testing process, 132

