# Index


**A**
Abstract data types (ADT), 353
Abstraction, 37–38, 207, 211
A/B testing, 503
Acceptance testing, 129
ACL2, 216
ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 301
Active reuse repository systems, 332
Activities and tasks, 290
Activity model, 364
Actor model, 354
Adaptation, 79–80, 104
Adaptive, 446
  case management (ACM), 30
  change, 226
  patterns, 476
  random testing (ART), 142
Adequacy criteria, 130
ADT, *see* Abstract data types (ADT)
Agenda, 31
Agenda-driven case management (adCM), 30
Agile
  development, 113
  iterative development method, 27
  Manifesto, 26
  methods, 26–27, 77
  process, 357
Agility, 512
AHP, *see* Analytic hierarchy process (AHP)
Algorithms, 286
All-Defs coverage, 136
All-Uses coverage, 136
Alternating variable method, 160
Analytic hierarchy process (AHP), 76
Anti-patterns, 312
APFD metric, 172
Application programming interface (API)
  evolution, 233–234
  stability, 35, 233
Application Service Providers (ASPs), 511
Application threat models, 448
Architecture
  analysis, 112
  configuration, 113
  drift, 118
  model, 111
  pattern, 100, 112
  styles, 99, 112
  Trade-Off Analysis Method, 103
Architecture-based adaptation, 475
Architecture-based self-protection (ABSP), 475
Architecture description languages (ADLs), 100–101, 111
Argumentation, 75–76
Ariane 5, 124
ART, *see* Adaptive random testing (ART)
Artifact recommendation, 390
Artificial intelligence (AI), 85
Aspect-oriented programming, 234, 235
ASPs, *see* Application Service Providers (ASPs)
Assets, 447
Assumptions, 54, 288
Assurance, 58, 59
Atomicity violations, 139
Automata, 174
Automated refactoring, 238–239




S. Cha et al. (eds.), *Handbook of Software Engineering*,
https://doi.org/10.1007/978-3-030-00262-6

Automated repair, 230
Automated test execution, 166–167
Automated test generation, 175
Automated testing, 179
Automatic programming, 370

**B**
Backlog, 29
Base-level subsystem, 474
Behavior-driven development (BDD), 176, 197
Big data analytics, 39–40, 46, 514
Binding time, 324
Bisimilar, 208
Blackbox, 465
Black box reuse, 323
Black box testing, 124, 129, 145
Blockchain technology, 45
Böhm-Jacopini theorem, 353
Booch method, 355
Boolean specification, 135
Bound, 145, 470
Boundary value analysis, 145
Bounded model checking, 212–213
Bounded verification, 470–472
Box-and-arrow charts, 13–14
Brainstorming, 61–62
Branch coverage, 132
Branch distance, 158
Büchi automata, 204
Bug detection, 229–230
Bug fixes, 228–229
Bugs, 125
Business process execution language (BPEL), 15–16
Business process management, 8
Business process modeling notation (BPMN), 15
Business process workflow architecture, 8

**C**
Call graph, 364
Capability Maturity Model (CMM), 23, 42, 356
Capability Maturity Model Integration (CMMI), 356
Capture and replay, 166
Card sorting, 62
Case study, 287, 291
Catalysis approach, 355
Category-partition method, 146
CCS, 207
Chaining approach, 161
Challenges, 287
Change
  comprehension, 268–269
  conflicts, 254
  decomposition, 251–252
  impact analysis, 263–264
  suggestion, 269–270
Characterizing set, 153
Chat bots, 393
Chromosome, 161
Class diagrams, 67, 68
Class model, 366
Client adaptation, 233
Cliff's $\delta$, 294
Clone detection, 261
Clone removal, 239
Cloud computing, 492, 493
CMM, *see* Capability Maturity Model (CMM)
Coalition, 8
Code
  coverage, 178
  decay, 224
  duplication, 231
  inspections, 247
  review, 247–256
  smells, 241–243
Cohen's $d$, 294
Cohesion, 353
Cohort, 290, 292
Collaboration environments, 376
Collaboration model, 366
Colored Petri Nets, 14
Combinatorial interaction testing, 148–149
Commonality, 342
Communication, 2
Competent programmer hypothesis, 138
Compilation, 286
Complex data structures, 467
Complexity, 286
Components, 101
Comprehension, 290
Computational tree logic, 205
Computation errors, 133
Conceptual integrity, 94, 115
Concolic execution, 164
Concurrency coverage, 140
Concurrency testing, 165
Condition coverage, 134
Configuration, 101
  management, 98, 101, 107
  management systems, 376
Conflicts of interests, 293
Conformance testing, 152–154
Confounding factors, 292

Connectors, 99, 101, 103–104
Constraint solvers, 162
Containers, 507
Contexts, 288
Continuous integration (CI), 129, 165, 177, 384
Continuous simulation, 21–22
ContraVision, 63
Control flow graph, 259, 460
Control dependence graph, 157
Control flow model, 364–365
Controlled experiments, 287, 291
Coordination, 3
  costs, 376
  environment, 376
  Pyramid, 377, 378
  technology, 376, 377
  tools, 376
Copyleft, 504
Corrective change, 225
Correlation tests, 294
Cost-benefit analysis, 116–117
COTS, 78
Counter example, 154
Coupling, 353
Coupling effect, 138
Coverage criteria, 130
Creativity, 304
Creativity workshops, 63
Crosscutting concerns, 234
Crossover, 161
Cross-system porting, 231
Crowd programming, 392
CUTE, 198

**D**
DART, 198
Data-driven, 289
Data flow, 460
  analysis, 367
  coverage criteria, 136
  model, 363–364
  testing, 135
Data mining, 46
Data structures, 286
Deadlock, 139
Debugging, 290
Decentralization, 512
Deductive reasoning, 289
Defects, 125
Definition, 135
Def-use pair, 135
Delta Debugging (DD), 264
Dependability, 81–82
Dependencies, 241
Design decision, 108
Designing
  architectures, 113–115
  techniques, 94
Development, 290
Disjunctive normal form, 135
DoD Standard 2167 and 2167A, 7
Domain, 323
  errors, 134, 145
  properties, 54
  testing, 134, 145
Domain-independent design, 96–104
Domain-informed design, 96, 104
Domain-specific language (DSL), 359
Domain-specific software engineering, 106–107
Draco, 333
DSL, *see* Domain-specific language (DSL)
DSPL, *see* Dynamic Software Product Lines (DSPL)
Dynamic analysis of process specifications, 18–19
Dynamic program analysis, 446
Dynamic Software Product Lines (DSPL), 343
Dynamic symbolic execution, 164–165

**E**
Early-career professional developers, 305
Easy Approach to Requirements Syntax (EARS), 64–65
EC2, *see* Elastic Compute Cloud (EC2)
Economics of software engineering, 94
Economies of scale, 512
Ecosystems, 108
Effect sizes, 294
Elastic Compute Cloud (EC2), 499
Elasticity, 493
Elicitation, 57
Embedded systems, 173
Empirical discipline, 286
Empirical method, 288–289
Empirical software engineering, 287, 352
Empirical studies, 287
Encapsulation, 355
End-user development, 183
Energy consumption, 183
Entropy, 224
Environment, 53
Equivalence partitioning, 145
Equivalent mutants, 138
ER models, 354

Errors, 125
Event coverage, 141
Event flow graph (EFG), 141
Event-interaction coverage, 141
Evolution, 58, 290
Exception management, 37
Exemplars, 85–86
Experience, 304
Experiment, 291
Experimental protocols, 310
Experimentations, 288
Expertise recommendation, 390
Explanatory theories, 308
Exploit generation, 461
Exploits, 464, 481
Extreme programming (XP), 27, 357

**F**
Facet-based scheme, 332
Factorial design, 292
Failures, 125
False negatives, 451, 454, 481
False positives, 451, 454, 480
Fault, 125
Fault-based testing, 137
Fault tree analysis, 21, 41
Feature, 324
Feature model, 357
Feature-oriented programming (FOP), 235
Feature-Oriented Reuse Method (FORM), 107, 338
Feedback-directed random testing, 143
Finite state machine, 14, 150
First-order logic, 195, 199, 216
Fitness function, 156
Flaky tests, 181
Flowchart, 364
Floyd's program verification method, 214
FORM, *see* Feature-Oriented Reuse Method (FORM)
Formal methods, 352
Formal specification, 350
Formal verification, 446, 466
Fourth paradigm, 370
Freelancers, 305
Freelancers for hire, 306
Functional testing, 130
Fuzzing, 461, 465
Fuzz testing, 144

**G**
Generalisability, 288
Generation, 461, 481
Generative reuse, 332
Genetic algorithms, 161
GenVoca, 333
GNU Affero General Public License (AGPL), 505
GQM methodology, 297
Grammar-based testing, 144
Graphical user interfaces (GUIs), 140, 354
  model, 151
  ripping, 151
  testing, 140
Grid computing, 510
Grounded theories, 289, 290
Guidance and control, 4–5
GUIs, *see* Graphical user interfaces (GUIs)

**H**
Haystack, 110
Hierarchical Petri Nets, 14
Hill climbing, 159
Hoare logic, 196, 468
Hoare's proof system, 213
HOL, 216
Hosted bare metal, 498
Human factors, 286
Hybrid (cyber physical) systems, 212
Hypotheses, 288

**I**
*IDEF0,* 353
Idioms, 286
IEEE Transactions in Software Engineering, 301
i* modelling approach, 71
Incremental Commitment Model (ICM), 25
Indicative, 54
Inductive reasoning, 289
Infrastructure-as-a-Service (IaaS), 492
Inheritance, 355
Instant messaging systems, 377
Integration tests, 128
Intelligent development assistants (IDA), 393
Internal structure, 95
International ERCIM Workshop on Software Evolution, 301
International Workshop on Empirical Software Engineering in Practice (IWESEP), 301
Internet of Things (IoT), 352, 514
Interruption management, 389
Interviews, 61
Invisible action, 207
Invocation coverage, 141

*ioco* conformance, 154
ISO 9000, 23
IWESEP, *see* International Workshop on Empirical Software Engineering in Practice (IWESEP)

**J**
Jackson structured programming (JSP), 7
Jackson system development (JSD), 354
Joint Application Development (JAD), 61–62
Joint International Workshop on Principles of Software Evolution, 301
JSD, *see* Jackson system development (JSD)
JUnit, 127

**K**
KAOS, 69–72
Kernel MetaMetaModel (KM3), 360
Keyword-driven testing, 167
Kirchhoff's law, 361
KM3, *see* Kernel MetaMetaModel (KM3)
Koala, 106, 339
Kripke structure, 365
K-tuples, 136

**L**
Labeled transition system (LTS), 154, 365
Legacy tests, 182
LHDiff algorithm, 294
Linear temporal logic (LTL), 197, 203, 217
Little-JIL, 16
Load balancer, 503
Load testing, 182
Logical coverage criteria, 134–135

**M**
Machine learning, 178
Magnitude, 291
Maintenance, 290
Map-reduce, 111, 510
Mass customization, 341
Mass produced software components, 330
Mathematical models, 289
MDA, *see* Model-driven architecture (MDA)
Mealy machines, 150
Measurement, 292
Mechanical Turk, 306
Meta-heuristic, 156
Meta-level subsystem, 474
Metamorphic testing, 154–155
Meta-mutants, 138
Meta-Object Facility (MOF), 359
Metaprogramming, 8
Microservices architecture, 508
Minimal Cut Sets (MCSs), 21
Mining software repositories, 293–294
Mixed-method methodology, 289
Mixed-method research methodology, 289
Mocking, 127
"4+1" model, 99
Model-based testing, 149–151, 174–175
Model checking, 20, 41, 74, 154, 197, 208, 365, 466
Model differencing, 261
Model-driven architecture (MDA), 358
Model-driven engineering (MDE), 358
Modelling, 58, 194
Model transformation (ATL), 359
Model-View-Controller (MVC), 108
Modification-traversing tests, 169
Modified condition/decision coverage (MC/DC), 134, 173
Modular decomposition, 7
Modularity, 98, 241
  violations, 242
Module interconnection languages, 96–98
MOF, *see* Meta-Object Facility (MOF)
Monkey testing, 144
Moore machines, 151
MoSCoW, 65, 76
MPLs, *see* Multiple Product Lines (MPLs)
Multi-method, 289
Multi-method research methodology, 289
Multiple condition coverage, 134
Multiple Product Lines (MPLs), 344
Mutants, 137
Mutation, 161
  analysis, 178
  operators, 137
  score, 137
  testing, 137
Mutual exclusion problem, 200

**N**
NATO Software Engineering Conference, 286
Negative results, 308
Negotiation, 76
Network virtualization, 495
Nonfunctional testing, 130, 182–183
Normal design, 78
NQTHM, 216

**O**
OBDD, *see* Ordered binary decision diagrams (OBDD)
Object, 292
Object Management Group (OMG), 355
Object orientation, 352
Object-oriented analysis (OOA), 355
Object-oriented design (OOD), 105–106, 355
Object storage, 497
Observation, 62, 288
Office automation, 8, 43
Offline testing, 151
OMG, *see* Object Management Group (OMG)
Online testing, 151
Open-source software (OSS), 352, 504
Operational profile, 142
Optimistic SCM systems, 384
Ordered binary decision diagrams (OBDD), 209
Orthogonal arrays, 149
Output fault, 152
Outsourcing, 304

**P**
PaaS, *see* Platform-as-a-Service (PaaS)
PAD, 353
Pairwise combinations, 148
Paradigm(s), 349
  shifts, 377
  to coordination, 378
Partial order reduction, 210–211, 218
Participant, 292–293
Partition testing, 145–148
Path, 462
  condition, 162
  coverage, 133
Pattern Name
  "Idea Inspired by Experience," 314–315
  "Mixed-Method Style," 315
  "Prima Facie Evidence," 313–314
  "Tool Comparison," 313
Patterns, 287, 312
Perfective change, 226
Performance, 182, 291
Pessimistic SCM systems, 382, 383
Petri Nets, 14
Platform-as-a-Service (PaaS), 492
Platform-independent models (PIMs), 359
Platform-specific models (PSMs), 359
Polymorphism, 355
Popper, 310
Positivist research, 308
Post-mortem questionnaire, 292
PowerSim Studio, 22
Precision, 456, 462, 480, 481
Preventive change, 226–227
Principles, 286
Prioritisation, 77
Privacy, 83, 453, 461, 481, 482
Private clouds, 493, 506–507
Probabilistic systems, 212
Problem frames, 65
Process, 2
  acquisition, 16–18
  agents, 10–11
  algebra, 206–208
  analysis, 18
  artifacts, 10
  evolution, 23
  improvement, 4
  mining, 17
  model, 350
  performance, 9
  programming, 8
  simulator, 18
  specification, 3, 4, 9
Product families, 106–107
Product line approach, 107
Program dependence graph, 259–261
Program differencing, 256–261
Programming by demonstration, 245–246
Promise conference series, 310–311
Proof-carrying code (PCC), 469, 470
Protective wrappers, 476–477
Protocol analysis, 62
PSMs, *see* Platform-specific models (PSMs)
PVS, 216

**Q**
Qualitative research, 289
Quality gateway, 74
Quantitative relationships, 289
Quantitative research, 289
Quasi-experiment, 287, 291–292

**R**
Race conditions, 139
Radical design, 78
Random testing, 141
Rational unified process (RUP), 24
RE, *see* Requirements engineering (RE)
Real-time systems, 173, 198
Refactoring, 236, 252, 286
  impact, 240–241
  practices, 239–240
  reconstruction techniques, 252–254

validation, 267
Reference architecture, 112
Refutability, 289
Regression testing, 129, 165
Regression test selection, 169
Rejuvenation
pattern, 478
process, 478
Repeating patterns, 308
Repertory grids, 62
Replication experiment, 292
Repository, 311
REpresentational State Transfer (REST), 109–111
Representativeness, 304
Reproducibility, 293
Reproduction, 161
Requirements, 53
Requirements engineering (RE), 51
Research methods, 287
RESTful interface, 499
Reusable assets, 323
Reuse, 78–79, 321
Reuse failure model, 336
RiSE Reference Model (RiSE-RM), 336
Risks, 512
Runtime checking, 180
Runtime verification, 198, 216–217

**S**
SaaS, *see* Software-as-a-Service (SaaS)
SADT, *see* Structured analysis and design technique (SADT)
Sampling, 290
Satisfiability modulo theorem (SMT), 212
Satisfiability (SAT) solving, 198, 208, 212
SCADA, 104
Scalability, 451, 454, 464, 481, 482
Scales, 293
Scenarios, 68
Scientific method, 288
Scrum method, 29, 357
SDN, *see* Software-defined networking (SDN)
Search algorithm, 156
Search-based repair, 230
Search-based Software Product Lines, 344
Search-based testing, 156–162, 183
Security, 82–83, 446, 452, 453, 482, 483
Security architecture, 450
Selective mutation, 138
Self-coordination, 386
Self protection, 472
Sequence diagrams, 68, 69, 371
Service-level agreement (SLA), 506
Service-Oriented Systems, 78, 79
Shared editor systems, 384
Shewhart Cycle, 6
Simulation, 208
Simulink, 22
Sink, 458
SLA, *see* Service-level agreement (SLA)
Smalltalk-80, 354
Smart transactions, 45
SMT, *see* Satisfiability modulo theorem (SMT)
Socio technical dependencies, 375, 386
Software architecture, 94, 99, 108, 356
Software-as-a-Service (SaaS), 492
Software component, 113
Software Configuration Management (SCM), 383
Software connector, 113
Software crisis, 286
Software-defined networking (SDN), 495
Software design, 93
Software development, 286
environments, 375
processes, 286
Software economics, 505–507
Software ecosystems, 96
Software engineering, 93
engineering paradigms, 349–371
engineering research, 286
Software inspection, 247–256
Software library upgrade, 231
Software process, 2, 356
Software Process Workshop, 8
Software product lines (SPL), 78, 321, 357
Software rejuvenation, 478
Software synthesis, 218
Soundness, 480
Source, 458, 461
code, 155
transformation, 243–244
Specification, 54, 194, 466, 482, 483
SPIN, 211
Spiral model, 25
SPL, *see* Software product lines (SPL)
Sprints, 29
Staging environments, 508
Stakeholders, 52
State coverage, 151
State machine model, 365
State machines, 68–69
Statement coverage, 131
Static analysis of process specifications, 19–22
Static process analysis, 41

Static program analysis, 446
Statistically representative sets of participants, 304
Statistical model checking, 198
Statistical models, 289
Statistical techniques, 290
Storage virtualization, 495
Strategic dependency (SD) model, 72
Strategic rationale (SR) model, 72
Structural testing, 130, 155
Structured analysis and design technique (SADT), 7, 353
Structured programming, 349
Stubs, 127
Styles, 99, 287
Survey, 287, 290
Sustainability, 84–85
Symbolic, 462
Symbolic execution, 162–163, 461
Synchronization coverage, 140
Systematic editing, 246
Systematic literature review, 290–291
Systematic reuse, 323
Systems of Systems, 79
System testing, 128
SZZ algorithm, 293

**T**
Taint analysis, 458, 461
TDD, *see* Test-driven development (TDD)
Technological singularity, 369
Temporal logics, 467
Test
  analysis, 175–176, 178–179
  automation, 176–177
  case, 126, 127
  case prioritization, 171–172
  data, 180
  driven development (TDD), 28, 176, 184
  driver, 127
  frames, 146
  inputs, 180
  levels, 127
  model, 149
  oracle, 128, 175, 179–181
  selection, 169–171
  set, 126
  suite, 126
  suite minimization, 167–169
Theorem Proving, 468
Therac-25, 124
Thread interleavings, 165
Thread schedules, 165
Threats, 446
Threats to the validity, 288
Timed automata, 173
Time sharing, 510
Tournament selection, 161
Traceability, 80–81
Tradeoffs, 484
Transfer fault, 152
Transition coverage, 151
Transition-pair coverage, 151
Transparency, 293
Trap property, 154

**U**
Ultimately periodic, 209, 212
Unified Modeling Language (UML), 352, 355
Unique input/output (UIO) sequence, 153
Unit test generation, 161
Unit testing, 127, 176
Universal truth, 288
Unstructured interviews, 289
Utility computing, 510

**V**
Validation, 73
Variability, 342
Verification, 74–76, 482, 483
Verification of process, 4
Versioning support, 382
Virtualization, 492, 493
Virtual machines (VMs), 493
Volere, 56
Vulnerability, 447

**W**
Walkthroughs, 74
Waterfall Model, 7, 96
Whitebox, 465
White box reuse, 323
White box testing, 124, 129, 155
WinWin, 76
$W$-method, 153
Workflow, 2
  management, 8
  modeling systems, 384
Workspace awareness, 389, 390
Wright, 103

**X**
$x$Unit, 127

**Y**
Yet Another Workflow Language, 16