

# INDEX

---

- 80:20 rule, 36, 325, 353
- acceptance testing, *see* testing sub-phases
- accident, 23, 38, 89, 142, 268–283
  - analysis, 278
  - damage control, 38, 278
  - definition, 268
  - prevention, 275–278
- accuracy
  - measurement, 324, 332
  - model, 360, 362, 363
- ad hoc testing, 75, 80, 89, 95, 104, 211
- ADR (active design review), 246
- ADT (abstract data type), 262
- agile development, 46, 231, 242
- ambiguity, *see* partition problem
- analysis tool
  - S-PLUS, 318, 381, 385
  - SAS, 319
  - SMERFS, 319, 386
- ANN (artificial neuron network), *see* risk identification techniques
- API (application program interface), 98, 385
- assignment
  - data, *see* data, D-operation
  - statement, 260
- ATAC, *see* test automation
- availability, *see* quality attribute
- AWK, 385
- axiom, *see* formal verification
- axiomatic correctness/proof, *see* formal verification
- background knowledge, *see* pre-requisite
- basic Musa model, *see* SRGM
- BBT (black-box testing), *see* functional testing
- beta testing, 29, 47, 101, 208–209
- block, 193, 256
- book organization, 6
- boundary
  - closed, 129
  - definition, 128–129
  - linear, 129
  - loop, 184
  - open, 129
  - other boundaries, 141–144
    - buffer, 142
    - capacity, 141
    - data, 141
    - dynamic, 141
    - loop, 141
    - output, 141
    - queue, 142
  - point, 129
- boundary problems, 131–132
  - boundary shift, 131
  - boundary tilt, 131
  - changing closure, 141
  - closure, 130, 131
  - detected by EPC, 133–134

- detected by Weak1  $\times$  1, 139–140
  - detected by Weak $N$   $\times$  1, 136–139
  - extra boundary, 131
  - missing boundary, 131
  - non linear boundary change, 141
- boundary testing, 36, 72, 127–146, 156, 184, 185, 197
  - approximate strategies, 140
  - definitions, 128–129
  - EPC (extreme point combination), 132–134
  - other boundaries, 141–144
  - points, *see* test point
  - problems, *see* boundary problems
  - queuing, 142
  - strong strategies, 140
  - Weak 1  $\times$  1, 139–140
  - Weak  $N$   $\times$  1, 135–139
- branching statement, *see* conditional statement
- Brown-Lipow model, *see* IDRM
- bug, 20
  
- C, 33, 205, 206, 230, 385
- C++, 205
- causal analysis, 34, 224, 293, 344–345, 368
- CBSE, 97, 206, 210, 229, 387
- cellular communication system, 134, 137, 150, 162
- CFG (control flow graph), 176
  - construction, 178–180
  - decision node, 176
  - junction node, 176
  - link, 176
  - loop, *see* loop
  - node, 176
  - path, 177
  - processing node, 176
  - segment, 177
- CFT (control flow testing), 36, 72, 176–186
  - concatenation, 180
  - concept, 176
  - execution, 182
  - loop, *see* loop testing
  - model construction, 178–180
  - model decomposition, 180
  - nesting, 180
  - oracle, 182
  - path combination, 180
  - path selection, 180–181
  - path sensitization, 181–182
  - usage, 186
- chapter dependency, 9
- chapter overview, 6
- checklist, 36, 72
  - artifact based, 247
  - basic types of, 104
  - black-box, 104
  - combined, 105
  - component, 104
  - feature, 104
  - hierarchical, 105
  - implicit, 104
  - inspection, 243, 247
  - limitations, 106
    - complex interactions, 106
    - coverage, 106
    - holes, 106
    - redundancy, 106
  - property based, 247
  - specification, 104
  - standards, 104
  - use of, 104
  - white-box, 104
- checklist-based testing, 36, 75, 86, 103–107, 207, 209
- Cleanroom, 13, 32, 162, 227, 229
- CMM (capability maturity model), 13, 232, 325
- CMM/CMML/P-CMM/SA-CMM, 233
- CMVC, *see* measurement tool
- code reading, 244–246, 248, 261
  - stepwise abstraction, 245
- combinatorial explosion, 182, 200
- compiler, 98, 244
- complexity, 33, 36, 37, 241, 242
- component testing, *see* testing sub-phases
- concatenation, 183, 255
- conditional statement
  - cascading if's, 179
  - if, 190, 255
  - if-then, 178, 193, 195, 255, 257
  - if-then-else, 178, 255, 256
  - switch-case, 178, 190
- contradiction, *see* partition problem
- correctness, 19–24, 251–257, 259–265
  - verification/proof, *see* formal verification
- cost
  - comparison, *see* QA comparison, cost
  - defect fixing, 29, 34
  - poor quality, 31
- COTS, 206, 210, 229
- coupling, 17, 38, 247, 278, 309
- coverage-based testing, 81–82
  - checklist coverage, *see* checklist-based testing
  - data coverage, *see* DFT
  - decision coverage, *see* partition coverage testing
  - path coverage, *see* CFT
- crash, *see* failure
- CVS, *see* measurement tool
  
- damage control, *see* accident
- data
  - C-use, 187, 190
  - D-D relation, 187
  - D-operation, 187
  - D-U relation, 187
  - dependency, 187

- items (variables and constants), 175, 185, 187–200
  - operation, 187
  - P-use, 187, 190
  - relation, 187
  - U-D relation, 188
  - U-operation, 187
  - U-U relation, 187
- data dependency analysis, 188
- data slice, 187, 189, 190, 194–197
  - coverage testing, 195–198
  - fan-shape, 198
  - frequently used, 199
  - important, 199
  - sensitization, 195
- DDG (data dependency graph), 188–195
  - characteristics, 191
  - construction, 189–195
    - backward data resolution, 191
    - forward data tracing, 191
    - indirectly via CFG, 192–194
    - information source, 191
    - procedure, 191
  - data selector node, 190
  - input node, 189
  - intermediate node, 189
  - loops, 194
  - output node, 189
  - storage node, 189
  - unconnected node, 192
- dead (DDG) node, 192
- dead (DDG) sub-graph, 192
- dead state, *see* unreachable state
- debug, 20
- debugger, 76, 97, 205, 246
- decision (other than programming), *see* quality engineering
- decision (programming)
  - combination, 111
  - node, 110
  - path, 110
  - predicate, *see* predicate
  - predicate testing, *see* predicate testing
  - table, 110
  - tree, 110, 120
- decision tree modeling, *see* TBM
- defect, 20–24
  - absence, 251
  - accident, *see* accident
  - bug, *see* bug
  - classification, 345–348
  - definition, 20
  - error, *see* error
  - error source, *see* error
  - fault, *see* fault
  - hazard, *see* hazard
  - location, 342
  - measurement, 304, 306
  - post-release, 340
  - pre-release, 340
  - presence, 251
  - relation, 22
  - tracking, *see* defect handling
  - type, 341
- defect analysis, 339–351
  - classified, 348–350
  - defect dynamics model, 327, 343–344
  - density, 324, 325, 330, 332
  - distribution, 340–343
  - general types, 339–340
  - trend, 343–344
- defect containment, 37–39, 267–283
- defect handling, 41–43, 306, 308
- defect injection test, 211
- defect prevention, 31–33, 39, 223–236
  - education and training, 225–228
  - error blocking, 224
  - error source removal, 224
  - formal method, 229
  - methodology-based, 230
  - process-based, 231–234
  - standards-based, 230
  - technology-based, 229
  - tool-based, 230–231
- defect reduction, 34–37, 39
- defect-free, 68, 232
- desk check, *see* inspection
- DFT (data flow testing), 72, 186–200
  - application, 198–200
  - coverage testing, 195–198
  - data slice, *see* data slice
  - model construction, 189–195
  - synchronization testing, 199–200
- doing the right things, *see* validation
- doing the things right, *see* verification
- domain testing, *see* boundary testing
- DRM, *see* quality model
- dynamic analysis, 213
  - ETA, *see* ETA (event-tree analysis)
  - prototyping, *see* prototyping
  - sequencing, 214
  - simulation, *see* simulation
  - timing, 214
- EF (experience factory), 234
- equivalence class, 107–110, 112, 125, 126, 131, 136, 144, 149, 154, 156, 181, 184
- error
  - blocking, *see* defect prevention
  - definition, 20
  - source, 20
  - source removal, *see* defect prevention
- ETA (event-tree analysis), 214, 277
- execution
  - symbolic, *see* symbolic execution
  - test, *see* test execution
- Fagan inspection, 34, 239–243, 247, 248, 275

- observations and findings, 241
  - participants, 240
  - process, 239
- failure
  - analysis, *see* defect analysis
  - catastrophic, *see* accident
  - containment, *see* defect containment
  - crash, 75
  - definition, 20
  - determination, *see* test oracle
  - impact, *see* ODC
  - rate, *see* reliability
- fat software, 5
- fault
  - definition, 20
  - tolerance, *see* fault tolerance
- fault tolerance, 5, 37, 267–275
  - backup, 269
  - checkpoint-and-recovery, *see* recovery blocks
  - duplication, 269
  - failure independence, 268
  - NVP, *see* NVP (n-version programming)
  - rare event, 268
  - recovery blocks, *see* recovery blocks
- flat operational profile, *see* Musa's OP
- flow chart, 150, 178, 179, 189, 192
- formal method, 31–33, 50, 225, 251, 281
- formal specification, 32, 252
  - algebraic, 262
  - descriptive, 252
  - logical, 253
  - operational, 253
  - post-condition, 254
  - pre-condition, 254
- formal verification, 32, 48–50, 229, 251–266, 278, 281
  - algebraic, 262
  - axiomatic, 32, 253–259
    - assignment schema, 256
    - schema, 254
  - functional, 33, 253, 260
  - model checking, *see* model checking
  - wp (weakest pre-condition), 253, 260
  - wp (weakest precondition), 33
- FORTRAN, 205, 206
- FSM (finite-state machine), 36, 72, 148–152, 253, 262
  - deterministic, 149
  - dynamic element, 149
  - input, 149
  - Mealy model, 149
  - Moore model, 149
  - output, 149
  - probabilistic, *see* Markov chain, 149
  - representation, 151–152
    - graphical, 149, 151
    - list, 152
    - tabular, 152
  - state, 149
    - static element, 149
    - transition, 149
- FSM testing, 153–160
- FTA (fault-tree analysis), 275–277
- functional correctness, *see* formal verification
- functional testing, 35, 74–76, 216
- Gilb inspection, 243, 308
- Goel-Okumoto model, *see* SRGM
- goto, 180, 182, 183
  - goto considered harmful, 178
- GQM (goal-question-metric paradigm), 234, 333
- graph (for testing)
  - CFG, *see* CFG
  - DDG, *see* DDG
  - entry node, 176
  - exit node, 176
  - final state, 150
  - initial state, 150
  - inlink, 176
  - outlink, 176
- GUI (graphical user interface), 17, 122, 213, 215, 288
- hazard, 38, 268, 275, 293
  - indices, 277
- hazard analysis, 275–277
  - other techniques, 277
- hazard function, *see* reliability
- hazard resolution, 278
  - control, 38, 278
  - elimination, 38, 278
  - reduction, 38, 278
- heterogeneous system, 279–281
  - modeling, 279–280
- hole (in checklist), *see* partition problem
- HTML, 113, 157, 159, 160, 168, 215
- IDRM (input domain reliability model), 372, 374–377
  - application, 376–377
  - Brown-Lipow model, 376
  - Nelson model, 375, 376, 380, 381
- IDSS, *see* measurement tool
- IEEE Standard 610.12, 20
- information hiding, 33, 206, 226
- input, 87, 88, 90, 92–94, 99
  - domain, 128, 129
  - space, 128
  - sub-domain, 129
  - vector, 128
- input domain partition
  - definition, 128–129
  - domain analysis, 128, 130, 132
  - problems, 130–132
  - testing, 130–132
- input domain testing, *see* boundary testing
- inspection, 5, 237–250
  - ADR, *see* ADR (active design review)

- checklist-based, 247
- code reading, *see* code reading
- desk check, 244
- detection technique, 247–248
- effectiveness, 248–249
- Fagan, *see* Fagan inspection
- Gilb, *see* Gilb inspection
- meetingless, 242
- phased, 243
- review, 244
- static analysis, *see* static analysis
- two-person, 242
- walkthrough, 244
- integrated testing techniques, 214–217
- integration testing, *see* testing sub-phases
- interaction (software), 22, 32, 89, 91, 106, 126, 145, 147, 149, 157, 172, 175, 205, 293
- interaction in CCSCC, 280
- interaction testing, 175–202, 206–208, 212
- interface, *see* interaction
- Internet, 3, 46, 142, 160, 167, 209, 212, 230
- ISO 9000, 232
- ISO-9126, 18
- IV&V (independent verification and validation), 16, 78, 97, 206, 210
  
- Java, 159, 160, 205, 215
- Jelinski-Moranda model, *see* SRGM
  
- large software systems, 4, 372
- lean software, 5
- legacy product, 211
- link, *see* graph
- LOC, *see* quality measurement
- loop, 177, 182–184
  - combination, 184
  - combinatorial explosion, 182, 200
  - concatenation, 184
  - deterministic, 183
  - do-loop, 183
  - for-loop, 183
  - nesting, 184
  - non-deterministic, 183
  - repeat-until-loop, 183
  - verification, *see* formal verification
  - while-loop, 183, 195, 256, 257
- loop testing, 182–186
  
- maintenance process, 39, 51, 64, 264, 286, 305, 309, 316, 323, 330, 350
- Markov chain, 72, 160–162
  - definition, 161
  - memoryless property, 161
  - state transition probability, 161
- McCabe, *see* quality measurement
- McCabe Test, *see* test automation
- Mealy model, *see* FSM
- meaning, *see* semantics
  
- measurement, *see* quality measurement
- measurement tool
  - CMVC, 43, 230, 318
  - CVS, 230
  - IDSS, 43, 318
  - REFINE, 318
- memoryless property, *see* Markov chain
- model checking, 33, 261
- Moore model, *see* FSM
- MTBF, *see* reliability
- Musa's OP (operational profile), 111–125
  - case study, 121–125
  - construction, 115–125
    - Musa-1, 117–119
    - Musa-2, 119–121
  - graphical, 120
  - implicit, 119, 120, 123
  - participants, 117
    - customer, 117
    - high-level designers, 117
    - manager, 122
    - planning and marketing personnel, 117
    - system engineers, 117
    - test planners, 117
    - tester, 122
  - tree-structured, 120
- Musa-Okumoto model, *see* SRGM
- mutation testing, 211
  
- Nelson model, *see* IDRM
- nesting, 154, 155, 157, 178, 180–183
- NHPP, *see* SRGM
- node, *see* graph
- NVP (n-version programming), 272–275
  - diversity, 273
  - independency, 273
  - other applications, 274
  
- ODC (orthogonal defect classification), 248, 327, 334, 345–350
  - analysis, 348–350
  - attributes, 345–346
  - causal analysis, 346
  - concept, 345
  - failure view, 345
  - fault view, 346
  - multi-way analysis, 365–368
  - one-way analysis, 348–349
  - risk identification, 365–368
  - two-way analysis, 349–350
  - web error, 347–348
- OO (object-oriented development/technology), 13, 157, 206, 210, 230, 245, 249
- OP (operational profile), 104, 111, 160, 207, 216, 373, 375, 376, 380, 387
  - benefit, 114
  - construction, 115
  - Markov, *see* Markov chain
  - Musa, *see* Musa's OP

- usage, 114
- open-source development, 46, 230
- output, 87, 88
  - analysis, *see* quality analysis and assessment
- over-defined input, *see* partition problem
- parallel conditional assignment, 190
- partition, 107, 128
  - definition, 108
  - problem
    - ambiguity, 130
    - contradiction, 130
- partition coverage testing, 107–111
- partition-based testing, *see* partition coverage testing
- PASCAL, 205, 206
- path, *see* CFG
- path testing, *see* CFT
- people, 95–97
  - (code) reader, 244–246
  - consumer, 22, 23
  - customer, 16, 17, 19
  - designer, 95
  - developer, 95
  - in inspection
    - author, 240–242, 245, 246, 248
    - author-inspector pair, 242
    - inspector, 237–248
    - moderator, 240, 241
  - OP constructor, *see* Musa's OP, participants
  - programmer, 95
  - test manager, 96
  - tester, 95
    - career path, 95
    - IV&V, *see* IV&V
    - programmer as, 95
    - team, *see* testing team
    - third-party, 97
    - user as, 95, 96
  - user, 15–20
    - extension to non-human user, 17
- Petri-net, 200, 262
- pre-requisite, 11
  - computer science, 12
  - mathematics, 11
  - software engineering, 12
  - statistics, 11
- predicate, 109, 110, 254, 255, 260
- predicate testing, 109–111
- predicate transform, *see* formal verification
- prime program, 180
- probability, *see* OP
- program calculus, *see* formal verification
- program comprehension, 245
- program decomposition, 180
- prototyping, 36, 91, 213, 218, 368
- PSC (prescriptive specification checking), 281
- purification level, 384
- Putnam model, *see* quality model
- QA (quality assurance), 3–5, 8, 24, 27–39
  - activities, *see* QA activities
  - alternatives, *see* QA alternatives and techniques
  - classification, 27–31
  - strategy, 54–56, 58–59
  - techniques, *see* QA alternatives and techniques
- QA alternatives and techniques
  - comparison, *see* QA comparison
  - defect prevention, *see* defect prevention
  - failure containment, *see* failure containment
  - fault tolerance, *see* fault tolerance
  - formal verification, *see* formal verification
  - inspection, *see* inspection
  - testing, *see* testing
- QA comparison, 285–299
  - applicability, 289–291
  - cost, 287, 295–297
  - effectiveness, 286, 291–295
  - environment, 285
  - questions, 285
  - summary, 297
- QIP (quality improvement paradigm), 55, 234
- quality, 3, 15–26
  - analysis, *see* quality analysis and assessment
  - aspect, *see* quality aspect
  - assessment, *see* quality assessment
  - assurance, *see* QA (quality assurance)
  - characteristics, 15
  - engineering, *see* quality engineering
  - evaluation, *see* quality evaluation
  - expectation, 3, 15–17
  - feedback, *see* quality feedback loop
  - framework, 18
  - goal setting, *see* quality engineering
  - historical perspective, 24
  - improvement, *see* quality improvement
  - management, *see* quality management
  - measurement, *see* quality measurement
  - modeling, *see* quality modeling
  - perspective, *see* quality view
  - planning, *see* quality engineering
  - prediction, *see* quality prediction
  - service, 25
  - view, *see* quality view
- quality analysis and assessment, 59, 301, 303, 306–312
  - data granularity, 305
  - explicit, 311
  - failure-based, 310
  - fault-based, 310
  - implicit, 309
  - indirect, 310
  - model, *see* quality model
  - reliability assessment, 310

- safety, 310
- quality assurance, *see* QA (quality assurance)
- quality attribute, 18, 19
  - availability, 37, 268
  - correctness, *see* correctness
  - CUPRIMDS, 19
  - dependability, 37, 132, 268, 279, 290
  - ISO-9126, *see* ISO-9126
  - sub-attributes, *see* ISO-9126
- quality characteristics, 19
- quality engineering, 3, 5, 24, 53–64
  - activities, 60
  - analysis and follow-up, 4
  - effort, 61
  - execution, 4
  - feedback, *see* quality feedback loop
  - planning, 4
- quality engineering process, 5, 313, 314
  - further refined, 316–317
  - refined, 303
- quality feedback loop, 303–321
  - analyses, 309–313
  - analyses and modeling, 312–313
  - external, 313
  - immediate, 308–309
  - implementation, 313–317
  - long-term, 313
  - management, 311–312
  - product release, 309–311
  - tool support, 317–319
- quality goals, 6, 53, 55, 68, 79, 80, 301, 309, 312, 313, 315, 333
  - defect density, *see* defect
  - reliability, *see* reliability
  - safety, *see* safety
  - setting, 54, 56–57
- quality improvement, 301, 337
- quality measurement, 4, 56, 59, 304–308
  - activity, 307
  - defect, *see* defect measurement
  - direct, 304, 306
  - environment, 306
  - in-process, 330
  - indirect, 304, 306–308
  - product internal, 307
    - LOC (line of code), 227
    - McCabe, 361
  - reliability, *see* reliability
  - result, 304, 306
  - safety, *see* safety
  - selection, 333–335
- quality model, 56, 58, 59, 323–337
  - comparison, 328–330
  - customization, 330
  - data requirement, 330–333
  - DRM (defect removal model), 327, 330, 334
  - generalization, 330
  - generalized, 324–327
  - dynamic, 326
  - overall, 325
  - segmented, 325
  - interconnection, 330
  - product-specific, 327–328
    - measurement-driven predictive, 328
    - observation-based, 328
    - semi-customized, 327
  - Putnam model, 326, 334, 344
  - selection, 333–335
- quality planning, 56–59
- quality quantification, 63, 301, 303, 320, 337
- quality view, 15, 56, 57
  - consumer, *see* quality view, external
  - customer, *see* quality view, external, 56
  - developer, *see* quality view, internal
  - external, 16, 17, 22
  - internal, 16, 22
  - manager, *see* quality view, internal
  - manufacturing, 16, 24
  - product, 16
  - tester, *see* quality view, internal
  - transcendental, 16
  - user, *see* quality view, external, 16, 56
  - value-based, 16
- quantifiable quality improvement, 8, 63, 64, 298, 301, 303, 304, 320
- queue, 142
  - batching, 143
  - capacity, 142
  - pre-emptive, 143
  - priority, 142
    - FCFS, 142
    - FIFO, 142
  - priority class, 142
  - synchronization, 143
- queuing testing, 142
- reader preparation, *see* pre-requisite
- reading, *see* code reading
- recovery block, 270–272
- REFINE, *see* measurement tool
- regression testing, 88, 92, 98, 205, 211
- relation, 108
  - reflexive, 109
  - symmetric, 109
  - transitive, 109
- reliability, 17–19, 23–25, 36, 39, 55–57, 60, 70, 72, 79–81, 111, 114, 159–164, 167, 216, 268, 305, 325, 340, 371
  - approaches, *see* SRE
  - assessment, 334
  - estimate, 375
  - failure rate, 375
  - hazard function, 378
  - improvement, 335, 384–385
  - MTBF (mean time between failure), 378
  - reliability function, 378
- result

- analysis, *see* quality analysis and assessment
- checking, *see* result checking and test oracle data, *see* quality measurement
- result checking, 68, 91, 93, 97, 131, 144, 153, 156, 166, 178, 185, 189
- review, *see* inspection
- right software, 4, 232
- risk, 325, 353
  - trees, 277
- risk identification, 29, 36, 216, 353–369
  - classified data, 365–368
- risk identification techniques, 353
  - comparison, 362–365
  - correlation analysis, 355
  - DA (discriminant analysis), 356–358
  - new, 356–362
  - NN (neural network), 358–359
  - OSR (optimal set reduction), 362
  - pattern matching, 362
  - PCA (principal component analysis), 356–358
  - regression model, 355
  - TBM, *see* TBM (tree-based modeling)
  - traditional, 355–356
- run, *see* test
- S-PLUS, *see* analysis tool
- S-shaped model, *see* SRGM
- S-TCAT, *see* test automation
- safety, 23, 38, 39, 56, 269, 275–279, 297
- safety assurance, 23, 45, 47, 50
  - accident prevention, *see* accident prevention
  - ETA, *see* ETA (event-tree analysis)
  - FTA, *see* FTA (fault-tree analysis)
  - hazard analysis, *see* hazard analysis
  - hazard resolution, *see* hazard resolution
- safety-critical software/systems, *see* software, safety-critical
- SAS, *see* analysis tool
- scenario-based testing, 372
- semantics, 246, 260
- sensitization, 87, 110, 142, 148, 153, 156, 166, 179, 181, 182, 185, 186, 189, 195, 196, 198
- silver bullet, 5
- simulation, 36, 91, 213, 218
- simulator, 91, 213
- slice, *see* data slice
- Smalltalk, 205
- SMERFS, *see* analysis tool
- software
  - auxiliary, 56, 325
  - aviation, 5, 121
  - business, 325
  - commercial, 4, 56, 98, 99, 116, 308, 318, 325, 334, 354, 359, 360, 386
  - defense, 5, 97, 116, 233
  - embedded, 16, 89, 207, 212, 267, 269, 270, 275, 276, 279, 281
  - government, 97, 116, 118, 353, 354
  - in heterogeneous systems, 91, 96, 207, 212, 279
  - menu-driven, 156–158, 167, 172
  - operating systems, 4, 87, 116, 167, 263
  - quality, *see* quality
  - safety, *see* safety
  - safety-critical, 29, 36, 38, 57, 58, 89, 141, 213, 214, 246, 263, 264, 268, 325
  - telecommunication, 29, 87, 207, 212, 227, 267, 325, 355, 357, 379, 386
  - testing, *see* testing
- software maintenance, 211
  - adaptive, 211, 286
  - corrective, 211, 286
  - perfective, 211, 286
- software process, 43, 59, 70
  - BOOTSTRAP, 233
  - CMM, *see* CMM (capability maturity model)
  - conformance, 231
  - definition, 231
  - development process, 24, 29, 32, 34, 41, 43–46, 48, 50, 51, 59, 60, 62–64, 68, 167, 274, 305, 308, 310, 344, 365, 386, 387
  - improvement, 59, 94, 233, 234, 243, 248, 308, 324, 333
  - incremental, 32, 35, 44, 49, 60, 63, 167, 231
  - incremental process, 46
  - ISO 9000, *see* ISO 9000
  - iterative, 35, 44, 49, 60, 63, 167, 231
  - iterative process, 46
  - maintenance process, 43
  - maturity, 232
  - PSP (personal software process), 232
  - SPICE, 233
  - spiral, 35, 44, 49, 60, 63, 167, 231
  - spiral process, 46
  - waterfall, 44–45, 48, 60, 231, 232
- SRE (software reliability engineering), 340, 371–387
  - assumptions, 373–374
  - concept, 371
  - implementation, 385–386
  - input domain approach, 372
  - models, *see* SRGM and IDRM
  - time domain approach, 372
  - tools, 385
- SRGM (software reliability growth model), 328, 372, 377–380
  - application, 379–380
  - basic Musa model, 379
  - Goel-Okumoto model, 328, 334, 379
  - Jelinski-Moranda model, 378
  - Musa-Okumoto model, 379



- NHPP (non-homogeneous Poisson process), 378
  - failure rate function, 379
  - mean function, 379
- NHPP (non-homogeneous Poisson process) model, 378
- S-shaped model, 379
- state explosion, 262
- state transition, *see* FSM
- state transition probability, *see* Markov chain
- static analysis, 246
- statistical web testing, 167–171
- statistics, *see* quality model
- stochastic process, 161, 372, 378
  - Markov chain, *see* Markov chain
  - stationary, 165
- structural testing, 35, 71, 74, 76–77, 205, 216, 291
- symbolic execution, 33, 36, 246, 260, 261, 264
- synchronization, 143
- synchronization testing, *see* DFT
- syntax
  - checking, 215
  - error, 248
  - testing, 160, 215
- system testing, *see* testing sub-phases, 47
- T3, *see* test automation
- TBM (tree-based modeling), 328, 359–362
- TBRM (tree-based reliability model), 335, 371, 372, 380–385
  - application, 382
  - construction, 381–382
  - reliability improvement, 384–385
- test
  - case, 87
  - cases, *see* test preparation
  - failure determination, *see* test oracle
  - models, *see* testing techniques
  - oracle, *see* test oracle
  - result checking, *see* result checking and test oracle
  - run, 87, 106, 112, 114
  - script, 98
  - sensitization, *see* sensitization suite, *see* test preparation, 88
- test activities, 68–70, 73, 85–94
  - execution, 88, 90–92, 97–99
  - planning, 85–86
  - preparation, *see* test preparation
  - test analysis and follow-up, 93–94
- test analysis and follow-up, *see* test activities
- test automation, 85, 97–100
  - ATAC, 99
  - instrumentation, 99
  - McCabe Test, 99
  - S-TCAT, 99
  - T3, 98
  - TestWorks, 99
  - web-testing, *see* web-testing tools
- test execution, *see* test activities, 182
- test management, 85, 90, 95–97
- test measurement, 90, 92–93
- test oracle, 76, 91–92, 97, 100, 131, 144, 148, 153, 154, 182, 198, 214
- test point
  - boundary point, 129
  - definition, 128
  - exterior point, 129
  - interior point, 129
  - OFF, 135–141
  - ON, 135–141
  - vertex point, 129
- test preparation, 85–90
  - test cases, 86–89
    - systematic, *see* test techniques
  - test procedure, 89–90
  - test suite, 88–89
- testing, 5, 7, 65
  - concept, 67–71
  - exit criteria, 78–80
  - informal, 69, 104
  - management, *see* test management
  - object, 74
  - people, *see* people
  - performance, 213
  - process, 68, 69
  - questions, 71–74
  - record-and-replay, 88
  - regression, *see* regression testing
  - specialized, 213
  - stress, 213
  - technique, *see* testing techniques
  - tester, *see* people
  - transaction flow, 200
  - usability, *see* usability testing, 213, 218
- testing sub-phases, 35, 73, 75, 203–210
  - acceptance testing, 45, 47, 208, 377
  - beta testing, *see* beta testing
  - component testing, 206
  - integration testing, 206
  - system testing, 207, 372
  - unit testing, 205
- testing team, 95–96
  - horizontal model, 96
  - IV&V, *see* IV&V
  - mixed model, 96
  - vertical model, 96
- testing techniques, 72
  - black-box testing, *see* functional testing
  - coverage-based testing, *see* coverage-based testing
    - testing
      - defect diagnosis, 210
      - defect injection, 211
      - defect risk based, 211
      - defect-based, 211
      - functional testing, *see* functional testing
      - integration, 214

- models, 72, 86
  - checklist, *see* checklist-based testing
  - FSM, *see* FSM, Markov chain, and UMM
  - partition, *see* partition-based testing, boundary testing, and Musa's OP
- mutation, 211
- specialized, 210–214
- structural testing, *see* structural testing
- usage-based testing, *see* usage-based testing
- white-box testing, *see* structural testing
- tree-structured operational profile, *see* Musa's OP
- truth value, 178
- UBST (usage-based statistical testing), *see* usage-based testing
- UMM (unified Markov model), 160, 162–167, 216
  - construction, 164
  - coverage testing, 165
  - hierarchical testing, 166
  - implementation, 166
  - overall usage testing, 165
  - selective state testing, 165
  - selective transition testing, 165
  - usage testing, 164
  - web testing, 167–171
- UMM testing, 160–171
- under-defined input, *see* partition problem
- unit testing, *see* testing sub-phases
- unreachable state, 153
  - identification, 155
- usability, 17, 19, 215
- usability testing, 95, 101
- usage-based testing, 80–81, 111, 160
  - Markov chain, *see* Markov chain
  - Musa, *see* Musa's OP
  - OP, *see* OP
  - UMM, *see* UMM
- V-model, 48–50, 204, 205
- validation, 4, 46–47
- VDM, 254
- verification, 4, 47–48
  - formal, *see* formal verification
- walkthrough, *see* inspection
- WBT (white-box testing), *see* structural testing
- web logs, 216
  - access log, 168, 169, 171–173, 217, 348, 376
  - error log, 341, 348, 376
- web reliability, 376
- web testing
  - browser rendering, 215
  - form, 215
  - FSM, 150
  - FSM-based, 157–160
  - functional, 215
  - HTML syntax, 215
  - integrated strategy, 214–217
  - Java, 215
  - link checking, 215
  - load, 215
  - OP-based, 112
  - statistical, *see* statistical web testing
  - transaction, 215
  - usability, 215
  - usage
    - 2-stage model, 120
    - by file type, 112
- web usage
  - uneven distribution, 113
- web-testing tools
  - Doctor HTML, 215
  - Net Mechanic, 215
  - W3C Validator, 215
  - Weblint, 215
- wp (weakest pre-condition), *see* formal verification
- WWW (world wide web), 3, 142, 150, 157, 212, 341
- XP (extreme programming), 13, 46, 231, 242
- Z, 254
- zero defect, 25, 57
- zero defection, 25, 57