

INDEX

One of the great defects of English books printed in the last century is the want of an index.

— *Lafcadio Hearn a.k.a Koizumi Yakumo*

- 1+1 redundancy, 207
- 1xEV-DO, 207–208
- 2-way tests, *See* pairwise testing
- 3-way tests, 235
- 4-way tests, 235
- 80/20 ratio, 507
- 80/20 rule, *See* Pareto analysis
- AAA server, 208
- ability to perform, 553
- Abnormal (A), 114–115
- abstract syntax notation one (ASN.1), 201
- ACC, *See* acceptance criteria change
- acceptance
 - beta, 435
 - criteria, 451, 461
 - criteria change (ACC), 464
 - test engineer, 464
 - test execution, 463–464
 - test in XP, 466–467
 - test plan, 461–463
 - test report, 464–466
 - testing, 17, 450
- access
 - audit, 529
 - control, 529
 - terminal (AT), 208
- accident, *See* mishap
- accomplish, 508
- accounting management, 199
- accuracy, 453, 529, 531
- acoustic test, 176
- actions , 365
- active mode, 207
- activities performed, 553
- activities, tasks, and responsibilities (ATRs), *See* ATRs
- activity, 428
- actual behavior, 19
- actual outcome, 15
- adaptability, 533
- adapters, 185
- adaptive maintenance, 440–441
- adaptive random testing, 254–255
- adequacy
 - criteria, 47
 - criterion, 44
 - of testing, 42–45
 - properties, 47
- adequate, 47
- adjacent domain, 136, 142
- adjustable, 508
- Administrator, 401
- AETG, 260
- age, 429
- agile software development, 18, 22
- airlink, 208
- Albrecht, A. J. , 380
- all path coverage, 96–97
- all-combination testing, 235
- all-c-uses, 122
- all-c-uses/some-p-uses, 123
- all-defs , 121
- all-du-paths, 123
- all-pair testing, *See* pairwise testing
- all-p-uses, 122
- all-p-uses/some-c-uses, 123

- all-uses, 123
- analyzability, 533
- approximate solution, 144
- arrays, 227–228
- arrival and resolution of defects (ARD), 425
- ASN.1, *See* abstract syntax notation one
- ASP, 291
- aspect-oriented def-use, 131
- aspect-oriented programs, 131
- assertions, 61
- Assign State, 328
- Assigned, 410
- assumptions, 364
- asynchronous transfer mode (ATM), 335
- ATRs, 569
- attractiveness, 533
- authentication, 199, 208, 458
- author, 56
- authorization, 199, 208, 394
- automatable, 403
- automated teller machine (ATM), 313–315
- automation progress, 417
- automation test group, 498
- availability, 203, 456, 458

- B mark, 217
- backdoors, 204
- backup , 454
- bad fix , 418
- base station controller (BSC), *See* radio node controller
- base transceiver station (BTS), *See* radio node
- basic input-output system (BIOS), 175
- basic
 - interconnection tests, 197
 - model, 487–488
 - testing, 571
 - tests, 193, 194–196
- BAT, *See* business acceptance testing
- Bayonet project, 426–428
- Bazooka project, 420–423
- Beck, Kent, 73
- behavior tests, 197
- Beizer, B., 131, 110
- BERT, *See* bit error test
- beta release criteria, 436
- beta testing, 435–436
- big-bang, 173

- BIOS, *See* basic input-output system
- BIT, *See* built-in test
- bit error test (BERT), 196
- black-box
 - component testing, 185
 - testing, 20–21, 163
- Blocked, 349
- Boomerang, 442–444
- boot tests, 194
- boring and time consuming, 396
- bot, 459
- bottom-up, 171–173
- bound checker , 77
- boundary
 - inequalities, 141
 - tilt, 149, 152
 - value analysis (BVA), 246–248
 - value tests, 205
- brainstorm, 433
- branch coverage, 98–99, 127
- breakpoint, *See* debugger
- breakpoints debugger, *See* debugger
- BSMI mark, 217
- buffer allocation, 211
- buffers overflow, 60
- buffers underflow, 60
- build, 165
- built-in test (BIT), 186–187
- Burnstein, Ilene, 568
- business acceptance testing (BAT), 17, 450
- business goal, 461
- BVA, *See* boundary value analysis

- calendar time, 474
- candidate set, *See* adaptive random testing
- capability maturity model (CMM), 5, 547–548
- capability tests, 197
- capture the response, 399
- Card, David N., 432
- career path, 511–512
- carry through, 508
- categories, 256–257
- categorization of errors, 433
- category partition method (CPM), 256–258
- causal analysis , *See* defect causal analysis
- cause-and-effect diagram, *See* Ishikawa diagram
- CBA-IPI, 553
- CCB, *See* change control board

- CCC mark, 217
- CE mark, 216–217
- certification of COTS components, 188
- CFG, *See* control flow graph
- Chainsaw project, 430
- change control board (CCB), 178
- change request (CR), 54, 60
- changeability, 533
- characterization of test cycles , 362
- characterizing sequence (CS), 287–290
- characterizing set, 287
- check points, 563
- check-in request form , 166
- Chillarege, Ram, 9
- choice construction, 41
- choices, 256–257
- cleanroom software engineering, 19
- cleanup, 346, 397, 400
- CLI, *See* command line interface tests
- clock time, 474
- Closed, 410
- closed boundary, 142
- closed domain , 142
- closed inequality boundary, 147–150
- Closed State, 330
- closure error, 143, 150, 152
- CMM, *See* capability maturity model
- CMM appraisal framework (CAF), 554
- CMMI, 554–555
- CMM-SE, 555
- CMM-SS, 555
- CMM-SW, 555
- code auditor, 77
- code review, 54
- code review checklist, 58
- co-existence, 533
- coincidental correctness, 107, 261
- collateral damage, 71, 376, 418
- combinatorial testing, 259
- command line interface (CLI) tests, 196
- commercial off-the-shelf (COTS)
 - component, 186
- Commit State, 328
- Commitment, 560
- commitment to perform, 553
- common features, 553
- communication, 561
 - commonality, 529
 - systems tests, 196–197
- communicativeness, 529
- compatibility, 459
- compatibility matrix, 178
- compatibility tests, 209
- competent, 502
- competent programmer hypothesis, 68
- competitive edge, 454
- complete path, 121
- complete testing, 13
- completely specified, 279
- completeness, 452, 529, 459
- compliance
 - information, 329
 - marks, 216
 - statistics, 437
 - tests, 357
- component, 158, 174, 298–299
- component software, 174
- comprehensive, 502
- computation error, 135
- conciseness, 529
- confidentiality, 203, 458
- configuration
 - management, 199
 - management system, 80–81, 84, 551
 - maximum, 194
 - minimum, 194
 - tests, 209
- conformance testing, 273
- considerate, 502
- consistency, 514, 529
- constraints, 258
- construction fault, 35
- continual improvement, 536
- control dominating system, 266–267
- control flow
 - definition*, 20, 89
 - graph (CFG), 64, 89, 90–93
 - testing, 64, 89
- control testing process, 574
- controlled, 502, 506
- coordinated architecture, 294
- coordinating interview, 506–507
- corrective
 - action, 434
 - maintenance, 440–441
 - measures, 419
- correctness, 452, 523–524
- correctness of acceptance tests, 467
- cost, 557
- cost of testing, 574

- COTS, *See* commercial off-the-self
 - component
- coupling effect, 68
- coverage
 - definition*, 14
 - all path, 96–97
 - fault, 290
 - matrix, 321–322
 - metrics, 321
 - pairwise, 240
 - PCO, 313
 - predicate, 100–101
 - state, 276, 308
 - statement, 97–98
 - transition, 276, 308
- CPM, *See* category partition method
- CPPUnit, 73
- cpu time, 209
- cpu utilization, 211
- CR, *See* change request
- crash defects (CD), 424
- Create state, 345
- criteria
 - acceptance, 451, 461
 - adequacy, 47
 - beta release, 436
 - data flow testing, 121–124
 - dual-use variables, 226
 - entry, 368
 - exit, 366
 - FCS readiness, 437
 - GUI testing capability, 395
 - input variables , 225
 - McCall’s quality, 529
 - multiple-type variables, 226–227
 - output variables, 225–226
 - path selection, 89
 - path selection, 94–101
 - pricing , 395
 - quality, 6, 527–528
 - revert and extension, 365
 - system integration entry, 182
 - system integration exit, 182
 - test adequacy for GUI, 219
 - test development, 393
 - test execution, 394
 - test maintenance, 394
 - test management, 394
 - test results, 394
 - tool evaluation, 393
 - vendor qualification, 395
- Critical, 410, 502
- Crosby, Philip, 5, 431
- cross-functionality
 - groups, 409
 - tests, 204
- CSA mark, 216–217
- C-Tick mark, 217
- cumulative failure, 476
- CUPRIMDS, 461
- current vector, 280
- c-use, 117
- customer, 450, 463
- customer focus, 535
- CUTE, 79
- cyclomatic complexity, *See* McCabe
 - complexity measure
- daily build, 165
- DART, 79
- data commonality, 529
- data conversion, 453–454
- data conversion testing, 204
- data declarations, 296
- data dominating system, 265–266
- data flow
 - definition*, 20
 - analysis, 109–110
 - anomaly, 113–115
 - graph (DFG), 64, 116–119
 - testing, 64, 115–116, 127
 - testing criteria, 121–124
- data integrity, 453
- DAVE, 129
- DCA, *See* defect causal analysis
- dd (type1) anomaly, 113–115
- debriefing meeting, 511
- debuggers, 77
- debugging
 - definition*, 68, 77
 - backtracking, 69
 - brute force, 68
 - cause elimination, 69
 - goals, 569
 - heuristic, 69–70
- decision point, 91
- decision tables , 248–252
- Decline State, 330
- decrement process, 487
- decryption, 203

- def-clear, 122
- defeat a test, 39
- defect
 - definition*, 9–10
 - age, 442
 - causal analysis (DCA) , 431
 - count, 522
 - density, 522
 - dynamic model, 442
 - management, 561–562
 - prevention, 60, 432, 552, 576
 - priority, 410
 - removal efficiency (DRE), 441
 - report, 418
 - review meeting, 419
 - schema, 412
 - severity, 410–411
 - tracking system, 464
 - type, 429
 - arrival rate (DAR), 424
 - closed rate (DCR), 424
 - rejected rate (DRR), 424
- Define (d), 114–115
- Defined and referenced (R), 114–115
- Defined but not referenced (D), 114–115
- definition clear-path, 119
- definition of maturity, 554
- degraded node tests, 207
- Deleted state, 346
- Deming prize, 2
- Deming, W. Edwards, 2, 4, 5
- denial-of-service (DoS), 458
- dependability, 468
- dependable, 508
- Deprecated state, 347
- design fault, 35
- design verification test (DVT), 174
- determine the verdict, 399
- determined, 509
- deterministic, 279
- develop testing, 569
- development rework cost, 523
- development test group, 498
- diagnostic test, 175, 195–196
- digital signature, 453
- discipline, 554
- distance between programs, 47
- distinguishing sequence (DS), 284–287
- distributed architecture, 293
- DMADV, 579
- DMAIC, 579
- documentation, 174, 461
- documentation tests, 194, 215–216
- documented, 398
- documenters, 77
- domain
 - definition*, 136
 - error, 64, 136–137
 - testing, 64, 137–138
- dominance, 108
- dormant defect, 442
- DoS, *See* denial-of-service
- Draft state, 345
- DRE, *See* defect removal efficiency
- drive the test, 399
- DS-tree, 285, 288
- du (type3) anomaly, 114–115
- du-path, 121
- Duplicate, 410
- DVT, *See* design verification test
- dynamic analysis, 7
- dynamic unit testing, 53, 62–64
- EAP, 204
- easy to automate, 396
- EC, *See* engineering change
- Eclat, 76
- ECO, *See* engineering change order
- effective domains, 155
- effectiveness of test cases, 11
- efficacy of testing, 45
- efficiency, 524, 531
- efficient, 566–567
- EFSM, 302–307
- EGT system, 79
- electrical test, 176
- electromagnetic emission test , 175
- electrostatic discharge (ESD) test, 175
- element management systems (EMS) tests, 198–202
- EMC compliance, 216
- emission and immunity, *See* EMC compliance
- EMS, *See* element management system tests
- emulators, 77
- encryption, 203
- end-to-end testing, 163
- end-to-end validity, 183
- endurance, 184

- engineering change (EC), 329, 437
- engineering change order (ECO), 177–180
- enlarged domain, 148–149, 151–152
- entry criteria, 182
- entry node, 93–94, 118
- entry point, 88
- environmental test, 176
- equality boundary, 153–154
- equipment handling, 176
- equivalence class, 244–245, 256
- equivalence class partitioning , 244–246
- error
 - definition*, 9
 - guessing, 20, 255–256
 - tolerance, 529
- ESD, *See* electrostatic discharge test
- Estelle, 18, 130
- estimating, 559
- Ethernet loop back tests, 196
- evaluate software quality, 576
- evaluation, 562
- events, 291
- exact solution, 144
- exception handler routine, 61
- executed set, *See* adaptive random testing
- execution status of test (EST), 420
- execution efficiency, 529
- execution environment, 478
- execution time , 209, 473
- exhaustive testing, 39
- exit criteria, 182, 366
- exit node, 93–94, 118
- exit point, 88
- expandability, 529
- expectation, 513
- expectation dependent , 42
- expected outcome, 15
- exterior point, 145
- external evaluations, 553
- external test architectures, 291
- extreme point, 142
- eXtreme Programming (XP), 8, 18, 22, 71, 466, 481

- factorial, 19
- factors, 236–237
- factual approach, 537
- FAD, 410
- Fagan, Michael, 54
- Failed, 349

- failure, 9, 472–473
- failure containment, *See* degraded node
 - tests
- failure intensity, 476–478, 487
- false negative, 399, 453, 502
- false positive, 399, 453, 502
- fast learner, 509
- fault
 - definition*, 9, 472–473
 - coverage, 290
 - injection, 20, 68, 84
 - management, 199
 - seeding, 20, 45, 441–442, 447
 - simulation, 20
 - tolerance, 531, *See also* high availability
 - tests
 - based testing, 20
 - free program, 45
- FCC mark, 216–217
- FCS, *See* first customer shipment
- FCS blocker, 437
- FCS readiness criteria , 437
- feasible
 - definition*, 90
 - data flow, 126
 - entry-exit path , 98
 - path, 126, 135
- feature
 - definition*, 321
 - tests, 204
- Feigenbaum, Armand V., 3
- finite-state machine (FSM), 18, 270–271
- Finkelstein, A. C. W., 325
- first customer shipment (FCS), 437
- fishbone diagram, *See* Ishikawa diagram
- five-C characteristics, 502
- FLASH card, 194
- Flexibility, 525
- Ford, Henry, 3
- formal verification, 83
- frame relay (FR), 335
- FrAtm, 336
- FSM, *See* finite-state machine
- full polling, 213
- function, 222
- function as designed (FAD), *See* FAD
- function as designed count, 423
- function points, 380
- functional specification, 18, 56
- functional testing, 20–21, 64, 222–232

- functional validity, 183
- functionality
 - definition*, 531
 - tests, 193, 196–204
- functionally related, 230
- functions and training, 561
- fundamental theorem, 33
- fUnit, 73

- Galín, D., 544
- Gantt chart, 389
- Garvin, D. A., 5, 519
- generality, 529
- Get, 201
- GetBulk, 201
- GetNext, 201
- Gilb's technique, 521
- global c-use, 119
- glue, 185
- goal-oriented, 509
- goals, 363
- gold standard oracle, 253
- GOST-R certification, 217
- Gotel, O. C. Z., 325
- graphical user interface (GUI)
 - tests, 202
- green bin, 370
- GUI, *See* graphical user interface tests
- GUI testing capability criteria, 395
- guidelines for automation, 395–396

- HALT, *See* highly accelerated life tests
- Hamlet, R., 10
- hardware independence, 529
- hazards, 216
- High, 410
- high availability tests, 206–207
- high risk, 396
- high-level design, 56
- highly accelerated life tests, 177
- Hold, 410
- homogeneous state block, 285
- homogeneous vector, 280
- Howden, William E., 222, 253
- Humphrey, Watts S., 432
- hybrid analysis, 7
- hypothesis, 70

- ideal test, 32
- ideal test selection, 38

- IETF, 534
- impact, 429
- implementation under test (IUT), 291
- Implement State, 328
- implications, 108
- IMSL, 222
- in parameter order (IPO), 240–243
- inappropriate action, 36
- inappropriate path selection, 35
- in-circuit emulators, 77–78
- includes, 124
- incremental, 164
- independent, 398
- independent test group, 572
- infeasible, 90, 104, 126
- infeasible path, 107–108, 135
- inferred requirements, 335
- information sharing, 514
- in-house training, 512
- initial, 569
- initial path vector, 280
- initial state block, 285
- initial vector, 280
- inject faults, 107
- innovative, 509
- inopportune input, 13
- in-process metrics, 419, 446
- input
 - classifier, 136
 - condition, 244
 - data, 106
 - domain, 14, 19, 64, 135, 244–245, 252, 233
 - sequence, 279
 - vector, 101–102
- inspection, 54
- installability, 459–460, 533
- installation procedure, 215
- instrumentation, 529
- instrumentation code, 60
- integrate testing into software life cycle, 572–574
- integrated software management, 551
- integration of hardware/software,
 - See* software/hardware integration
- integration test group, 26, 496
- integration testing, 17, 158–159
- integrity, 203, 525
- integrity checking, 454
- intended behavior, 19

- interacting quality processes, 537
- interactive debuggers, *See* debuggers
- interface
 - errors, 160–162
 - integrity, 183
 - stress, 183
- intergroup coordination, 552
- interior point, 145
- inter-method testing, 130–131
- internal assessments, 553
- interoperability, 459, 526, 531
- interoperability tests, 193, 208–209
- interprocedural data flow testing, 131
- intersystem testing, 163
- interviewing, 507
- intraclass testing, 130–131
- intramethod testing, 130–131
- intrasystem testing, 163
- Invalid , 349
- invalid input, 13, 244
- invariants, 61
- involvement of people, 536
- IPD-CMM, 555
- IPO, *See* in parameter order
- IPSec, 204
- IRAM safety mark, 217
- Irreproducible, 410
- irreproducible defects (IRD), 424
- Ishikawa diagram, 4, 431, 433–434
- Ishikawa, Kaoru, 3–4, 433
- ISO
 - 900 principles, 535–537
 - 9001, 535
 - 9001 requirements, 537–541
 - 9126, 6, 521, 530–534
- IUT, *See* implementation under test

- job profiling, 505
- job requisition, 504–505
- Jones, Capers, 384
- junior engineers, 501
- JUnit, 73
- Juran, Joseph, 3, 431
- JUSE, *See* Union of Japanese Scientists and Engineers
- just in time, 377

- key areas, 558, 564–565
- key process areas (KPA), 549–550
- Kitchenham, B., 5

- Kohavi, Z., 288
- Koomen, T. and Pol, M., 556
- Korat, 76
- KPAs, *See* key process areas

- LDAP, 204
- lead time, 556–557
- leadership, 535
- lean principle, 2
- learnability, 521, 533
- LED, *See* light emitting diode tests
- less than perfect, 377, 461
- less volatile, 396
- levels, 237
- life cycle
 - model, 559
 - of a defect, 366–367, 409
 - of an individual requirement, 323–324
 - of a test case, 345
- light emitting diode tests, 195
- limitation of testing, 45–46
- Lipton, Dick, 65
- load and stability tests, 194
- load tests, 213
- local architectures, 291–292
- log the verdict, 400
- logarithmic model, 489
- logging and tracing tests, 198
- logic fault, 34
- loop-free path, 121
- LOTOS, 316
- Low, 410
- lower tester, 292–293
- low-level design, 56
- low-level testing, 562

- machine identification, *See* distinguishing
 - sequence
- maintainability, 457, 525, 531
- maintainable, 398
- maintenance, *See* software maintenance
- maintenance mode, 187
- Malcolm Baldrige National Quality Award, 5
- malleable, 508
- management and measurement, 575
- management information base (MIB) , 200, 202
- management requirements, 538
- manually difficult, 396

- manufacturing view, 6, 452, 520
- marketing beta, 435
- Martin, Robert C., 72
- MATLAB, 227
- matrix
 - coverage, 321–322
 - traceability, 325, 358
- maturity, 531
- maturity goals , 569
- maturity
 - level, 549–551
 - level 1: initial, 550
 - level 2: repeatable, 551
 - level 3: defined, 551
 - level 4: managed, 552
 - level 5: optimizing, 552
- maturity models, 546
- maturity subgoals, 569
- McCabe complexity measure, 79, 108
- McCall, J. A., 523, 527, 529
- McCall's quality
 - criteria, 529
 - factors, 524
- MD, *See* message digest
- mean number of failures, 487
- mean time between failures (MTBF), 177, 214, 456, 475
- mean time to failure (MTTF), 214, 475
- mean time to repair (MTTR), 214, 457, 475
- measurement and analysis, 553
- measuring quality, 521
- measuring test execution, 22–23
- Medium, 410
- memory
 - carving, 211
 - leak, 211
 - leak detectors, 78
 - usage, 209
 - utilization, 211
- mentoring, 512–513
- merge point, 91
- message digest (MD), 453
- message passing interface, 160
- meticulous, 508
- metrics
 - spoilage, 442–443
 - test, 559–560
 - test effectiveness, 441
 - code review, 60
 - coverage, 321
 - monitoring defects, 419
 - monitoring test execution, 419
 - test case design effectiveness, 350–351
 - test case effectiveness , 23
 - test design preparedness, 349–350
 - test effort effectiveness, 23
- MIB, *See* management information base
- MIC mark, 217
- milestones, 2, 53, 387
- Mills, H. D., 19, 447
- mishap , 216
- missing control-flow path, 35
- missing paths, 21
- mitigating action, 434
- modeling test results, 347–349
- moderator, 56
- modular, 397
- modularity, 529
- module, 158, 296
- module tests, 197–198
- moment of involvement, 559
- monitor testing process, 574
- monitoring test execution, 22–23
- Mothra, 68
- motivation, 560
- MTBF, *See* mean time between failures
- MTTF, *See* mean time to failure
- MTTR, *See* mean time to repair
- multiple-experiments, 288
- Musa, John D., 10, 20, 492–493
- mutant
 - definition*, 65
 - dead, 65
 - equivalent, 65
 - killable , 65
 - killed, 65
 - stubborn, 65
- mutation analysis, 20, 65
- mutation score , 65
- mutation testing, 65
- mutation adequate, 65
- mutually beneficial, 537
- Myers, G., 68
- NE, *See* network element
- network analyzers, 80
- network element (NE), 198
- network management station (NMS), 200
- New, 410
- next-state function, 277–278

- NMS, *See* network management station
- NOM mark, 217
- nondeterministic finite-state machine (NFSM), 316
- nonlinear predicate, 155
- nonterminal, 281, 285
- non-testable, 46
- normal mode, 187
- NTLM, 204
- numeric variable, 224
- NUnit, 73

- OA, *See* orthogonal array
- objective of testing, 10–11
- object-oriented programs, 51, 130
- observable failure, 207
- observation, 271
- observers, 57
- ODC, *See* orthogonal defect classification
- OEM, *See* original equipment manufacturer
- OFF point, 145
- office environment, 560
- off-the-shelf (OTS) component, 184–187
- Ohna, Taiichi, 2
- OIR, *See* on-line insertion and removal tests
- omniscient debugger (ODB), *See* debuggers
- ON point, 144
- on-line help, 215
- on-line insertion and removal tests, 206
- ON-OFF point, 155
- on-site commercial training, 512
- Open, 410
- open boundary, 142
- open domain, 142
- open inequality boundary, 150–153
- Open State, 326
- openfiles(), 91–92, 96
- operability, 521, 529, 533
- operation rework cost, 523
- operational profile, 10, 19, 457, 482–484
- operational system testing, 186
- optimization, 567, 576
- oracle
 - definition*, 12, 45, 253
 - gold standard, 253
 - parametric, 253
 - perfect, 253
 - statistical, 253–254
- organization process definition, 551
- organization process focus, 551
- original equipment manufacturer (OEM), 178, 435, 451
- orthogonal array (OA), 236–240
- orthogonal defect classification (ODC), 428–431
- open system interconnection (OSI), 291
- OTS, *See* off-the-shelf component
- output domain, 19, 64, 233
- output function, 277–278
- output sequence, 287–289
- outstanding defects (OD), 424

- packaging, 176
- packet data serving node (PDSN), 208
- pair programming, 73
- pairwise
 - coverage, 240
 - testing, 163–164, 235
 - validity, 183
- parametric oracle, 253
- Parasoft, 110
- Pareto analysis, 430–431, 433
- Pareto, Vilfredo, 431
- Parhami, Behrooz, 9
- partition testing, 260
- partition the input domain, 39
- Passed, 349
- path
 - definition*, 88–89, 135
 - predicate, 102, 142
 - predicate expression, 104
 - selection criteria, 89, 94–101
 - vector, 280
- PCMCIA card, 194
- PCO
 - definition*, 269
 - coverage, 313
- PDCA cycle, *See* Shewhart cycle
- PDU, *See* protocol data unit
- PEAP, 204
- peer review, 552
- perfective maintenance, 440–441
- perfect oracle, 253
- performance
 - characteristics, 455
 - fault, 35
 - management, 199
 - monitors, 79
 - test group, 498

- tests, 193, 209–210
- permanent virtual connection (PVC), 335
- perturbation
 - analysis, 84
 - technique, 155
 - testing, 83–84
- PhAge, *See* defect age
- phase definition, 569
- plan-do-check-act (PDCA) cycle,
 - See* Shewhart cycle
- planned versus actual execution (PAE), 420
- planning, 559
- point of control and observation (PCO),
 - See* PCO
- point-to-point protocol (PPP), 208, 238
- portability, 525–526, 531
- ports, 298–299
- POST, *See* power-on self test
- post-conditions, 61
- Postponed, 410
- power cycling tests, 206
- power-on self test (POST), 195–196
- precise, 508
- pre-conditions, 61
- predicate coverage, 100–101
- predicate interpretation, 103
- presenter, 56
- preventive
 - action, 52, 434
 - measures, 60, 419
- pricing criteria , 395
- principal engineers, 501
- prioritization
 - a test maturity matrix, 563
 - of requirements, 326
 - of test cases, 371–372
- probability of failure-free operation, 477
- procedure call interface, 159
- process approach, 536
- process change management, 552
- product
 - operation, 526–527
 - revision, 526–527
 - transition, 526–527
 - view, 6, 452, 520
- program
 - dependent, 41
 - inference, 47
 - mutation, 45
 - path, *See* path
 - adequate, 47
 - programming, 56
 - protocol data unit (PDU), 202, 291, 297–298
 - prototyping model, 481
 - PSB mark, 217
 - public forum training, 512
 - p-use, 117
 - push-to-talk, 209
 - PyUnit, 73
 - QA, *See* quality assurance
 - QC, *See* quality circle
 - QMS, *See* quality management system
 - Qualifier, 429
 - qualitative segments, 566
 - quality
 - assurance (QA), 499
 - attribute, 461
 - characteristics, 6, 468, 530–531
 - circle (QC), 4, 431
 - control , 500
 - controller, 500
 - criteria , 6, 527–528
 - evaluate software, 576
 - factors, 6, 523–528
 - management group, 499
 - management system (QMS), 537
 - McCall's criteria, 529
 - McCall's factors, 524
 - measuring, 521
 - metrics , 530
 - model, 6
 - objective, 537
 - of the system, 556
 - policy, 537
 - software, 5–7, 519, 530
 - subcharacteristics, 532
 - quantitative process management, 552
 - quick polling, 213
 - quick study, 509
 - radio access network (RAN), 208
 - radio node (RN), 207
 - radio node controller (RNC), 207
 - RADIUS, 204
 - random testing, 10, 127, 252–254
 - Rao, C. R., 236
 - RCA, *See* defect causal analysis
 - reaching definitions, 131

- read-only memory (ROM), 175, 194
- realization requirements, 540
- recognition, 515
- recordkeeper, 56
- recoverability, 531
- recovery, 454
- recruiting process, 504
- red bin , 369
- reduced, 279–280
- reduced domain, 147–148, 150–151
- reduced flow graph, 108
- Rees, Roderick, 9
- Reference (r), 114–115
- regression test, 17, 194, 214–215, 404–405, 418
- regulatory tests, 194, 216–218
- relationship building, 509
- release note, 166, 216
- Released state, 346–347
- reliability, 456, 471–472, 524, 531
- reliability models, 486–491
- reliability tests , 177, 194, 214
- reliable criterion, 33, 38
- remedial requirements , 541
- remote architecture, 295
- repeatability, 396
- repeatable process, 546
- repeatable results, 546
- replaceability, 533
- reporting, 561
- requirement
 - definition*, 56
 - engineering, 326
 - fault, 34
 - identification, 322–330
 - inferred, *See* inferred requirements
 - life-cycle, 323
 - management, 551
 - of a system, 36
 - schema , 324
 - traceability, 325
- reset sequence, 278
- Resolved, 410
- resource requirements , 539
- resource utilization, 209, 533
- response time, 209
- results driven, 509
- retaining testers, 511
- returnaverage(), 92, 97, 118
- reusability, 526
- reusable, 398
- revealing criterion, 39
- revert and extension criteria, 365
- Review State, 327, 346
- review
 - code, *See* code review
 - functional specification, 332
 - program, 575
 - requirement, 331–332
 - system test plan, 356
- reviewers, 56
- reward system, 513
- rework-cost, 522
- robust and reliable, 397
- robustness
 - tests, 193, 204–208
 - of a system, 458
- ROM, *See* read-only memory
- root cause
 - analysis (RCA), *See* defect causal analysis
 - defect analysis, *See* defect causal analysis
- runs, 237
- SA, *See* safety assurance
- SABS mark, 217
- safety assurance (SA), 218
- safety tests, 177
- scaffolding, 62
- scalability
 - of a system, 460
 - test group, 498
 - tests, 193, 210–211
- schedule, 574
- scheduling system testing, 387
- scope of methodology , 561
- screening resume, 505–506
- Scrum, 18, 481
- scrupulous, 508
- SDL, 18, 303
- secure shell (ssh), 401
- security
 - definition, 531
 - management, 199
 - tests, 203–204
- selection criteria
 - for dual-use variables, 226
 - for input variables, 225
 - for multiple-type variables, 226–227
 - for output variables, 225–226

- self-documentation, 529
- self-sufficient, 398
- senior engineers, 501
- sequential computation, 91
- service interworking, 336, 379, 390
- serviceability, 458
- Set, 201
- setup, 346, 397, 399
- shared memory interface, 159–160
- Shelved, 410
- Shewhart cycle, 2
- Shewhart, Walter Andrew, 3
- shifted-boundary error, 143
- shrink-wrap, 366
- silent fail-over , 207
- simple, 397
- simple network management protocol (SNMP), 200–202
- simple path, 120
- simplicity, 529
- simulators, 80
- singleton state block, 285
- singleton vector, 280
- SIT, *See* system integration testing
- Six Sigma Maturity Model, 579
- SNMP, *See* simple network management protocol
- soft handoffs, 212
- softer handoffs, 212
- software
 - acquisition, 554
 - configuration management, *See* configuration management system
 - efficiency, 529
 - engineering institute (SEI), 547
 - image, 165
 - inspection support, 78
 - maintenance, 440
 - module, *See* module
 - process, 546
 - product engineering, 552
 - project planning, 551
 - project tracking, 551
 - quality, 5–7, 519, 530
 - quality assurance (SQA), 551
 - quality assurance (SQA) group, 499
 - quality management, 552
 - reliability, 10, 477–480
 - subcontract management, 551
 - system independence, 529
 - hardware integration, 174
 - source, 429
 - special training, 512
 - specification dependent, 42
 - specification adequate , 47
 - SQA, *See* software quality assurance
 - SSL, 204
 - stability, 533
 - stability tests, 213
 - stages, 568
 - stakeholders, 10, 323, 409
 - standardization, 514
 - standby mode, 207
 - start-up time, 456
 - state coverage, 276, 308
 - state verification, 278–279
 - stateless systems, 11, 265
 - statement coverage, 97–98
 - state-oriented systems, 11, 265
 - state transition, 278
 - state transition sequence, 273
 - static analysis, 7
 - static code analyzer, 78
 - static test technique, 559
 - static unit testing, 53
 - statistical oracle, 253
 - statistical quality control (SQC), 2, 577
 - statistical testing, 493
 - Stinger project , 425–426
 - stop testing, 43
 - stories, 466–467
 - story, 71
 - stress tests, 193, 211–213
 - stress the system, 456
 - strictly includes, 124
 - strongly connected, 280
 - structural testing, 20–21
 - structure, 554
 - stubs, 62
 - subdomain, 39, 136, 244
 - Submit State, 326
 - subroutine arguments, 229
 - substructure, 229
 - subsystem, 158
 - suitability, 531
 - supplier, 464
 - sustaining
 - phase, 439–440
 - test engineers, 440
 - test group, 498

- Swanson, E. B., 440
- Symstra, 76
- Synergetic, 508
- system
 - definition*, 158
 - approach, 536
 - evolves, 325
 - integration group, 162
 - integration test plan, 180–184
 - integration testing (SIT), 158, 164–173
 - resolution tests, 197
 - test automation, 391–392
 - test execution, 408
 - test group, 26, 497
 - test plan, 356
 - test team, 500
 - testing (ST), 17
 - testing entry criteria, 368
 - under test (SUT), 269, 274
- systematic error, 433
- systematized, 508
- systemic requirements , 537
- system-level
 - fault injection testing, 185
 - testing, *See* system testing
- tactful, 509
- tactically, 73
- tactics, 509
- Taguchi, Genichi, 236
- Tailoring, 185
- Target, 429
- taxonomy of system tests, 192–193
- TCDY, *See* test case design yield
- TCE, *See* test case escaped
- TDD, *See* test driven development
- team building, 513–515
- team player, 508
- technical
 - beta, 435
 - leaders , 501
 - training program, 573
- technology change management, 552
- tenacious, 509
- terminal node, 281
- terminal state block, 285
- test (sub)group, 335
- test adequacy, *See* adequacy of testing
 - criteria for GUI, 219
- test approach, 357
- test architectures, 291–295
- test automation,
 - benefits, 24–26
 - design characteristics, 397
 - infrastructure, 400–401
 - practices, 401
 - structure, 399
 - tools, 392–395, 405
- test beds allocation, 416
- test case, 22, 300–302
- test case creation effort, 384–385
- test case dependency, 274
- test case design yield (TCDY), 350
- test case escapes, 420
- test case escaped (TCE), 23, 350, 352
- test case execution effort, 385
- test case library, 401
- test case life cycle, *See* life cycle of a test case
 - test case scheduler, 274
 - test case schema, 346
 - test case verdicts, 299–300
 - test case validation, 258
 - test case verification, 352
- test cases allocation, 416
- test coordinate procedures, 292–293
- test coordination, 15
- test coverage analyzer, 78
- test cycle, 362
- test data, 36, 230
- test data generator, 78–79
- test design, 22
- test design factors, 321
- test design process, 345–347
- test development criteria, 393
- test driven development (TDD), 22, 71
- test driver, 62
- test effort estimation, 377
- test environment, 358–361, 514–515, 560
- test execution, 347, 364
- test execution criteria, 394
- test execution rate, 417
- test execution strategy, 361
- test factory, 346
- test groups, 496
- test harness, 79
- test input, 101–106, 244–245
- test levels, 16–18
- test maintenance criteria, 394
- test management criteria, 394

- test managers, 500–501
- test maturity model (TMM), 6, 568–577
- test measurement program, 575
- test method, 42
- test milestones , 387
- test objective, 14, 22, 345
- test objective identification, 334–335
- test oracle. *See* oracle
- test plan, 21
- test plan for system integration, *See* system integration test plan
- test planning process, 570
- test platform, 400
- test predicate, 36
- test process improvement (TPI), 6, 555–567
- test process management, 562
- test process optimization, 577
- test purpose. *See* test objective
- test report, 16, 438–439
- test result schema, 348
- test result criteria, 394
- test selection criterion, 146–154
- test selection procedure, 369
- test space, 486
- test specification technique, 559
- test steps, 22, 346, 397
- test strategy, 558
- test suite
 - definition*, 335
 - FrAtm, 342
 - schema , 348
 - structure, 358
- test system, 274
- test team organization, 26
- test tools, 24–26, 560
- test vector, *See* test data
- test verdict, 15
- testability, 525, 533
- testable requirements, 331
- testbed, *See* test environment
- tester-developer ratio, 497
- test first, 71
- testing activities, 14–16
- testing and test control notation, *See* TTCN-3
- testing effort, 574
- testing system, 40
- testing theory, 31
- testware management, 562
- Theory of
 - Goodenough and Gerhart, 32–37
 - Gourlay, 39–42
 - Weyuker and Ostrand, 37–39
- thermal test, 176
- thorough, 508
- thorough test, 33
- throughput, 209
- tidy, 508
- tilted-boundary error, 143
- time behavior, 533
- timeliness, 458
- TLS, 204
- TMM, *See* test maturity model
- TMP, 294
- TMPDU, 294
- tool evaluation criteria, 393
- tools, 401
- top-down, 167–170
- total quality control (TQC), 3
- total quality management (TQM), 236
- TPI, *See* test process improvement
- Traceability, 529
- traffic generators, 80
- training, 512, 529
- training program, 551
- transcendental view, 5, 452, 519–520
- transfer sequence, 278
- transition coverage, 276, 308
- transition tour, 273–276
- Trap, 201
- Trigger, 429
- trivially valid, 38
- troubleshooting guide, 70, 216, 437
- trustworthy, 508
- TSL, 256, 258
- TTCN-3, 256, 295–302
- TTLS, 204
- types of interfaces, 159
- UAT, *See* user acceptance testing
- UIO sequence, 279–284
- UIO tree, 280–281
- UML, *See* unified modeling language
- Undefine (u), 114–115
- Undefined (U), 114–115
- Understandability, 521, 533
- unified modeling language (UML), 481
- Union of Japanese Scientists and Engineers (JUSE), 2

- unique input/output (UIO) sequence,
 - See* UIO sequence
- unit testing, 16, 51–53
- Untested, 348–349
- Updated state, 346–347
- Upgradability, 459–460
- upgrade/downgrade tests, 195
- upper tester, 292–293
- ur (type2) anomaly, 114–115
- usability, 455, 521, 525, 531
- usability testing, 202
- usage profile, *See* operational profile
- use space, 486
- user, 450
- user acceptance testing (UAT), 17, 450
- user manuals, 215
- user view, 6, 452, 520

- valid criterion, 33, 38
- valid input, 13, 244
- validation, 7–8
- value-based view, 6, 452, 520–521
- VCCI mark, 217
- vector perturbation, 280
- vendor qualification criteria, 395
- verbal communication, 509
- verification, 7–8
- verification sequence, 278
- Verification State, 329
- verifying implementation, 553

- version , 165
- version control, 80–81
- Vinter, D., 322
- Virus, 459
- Visionary, 509
- V-model, 16

- Wait, 410
- waiting time, 209
- walkthrough, 54
- Waterfall model, 18, 481
- web-based application, 461
- white bin, 370
- white-box testing, 20–21, 163
- wireless data network, 207–208
- w-method, 287
- worm, 459
- wrapper, 185
- written communication, 508
- w-set, 287–288

- XP, *See* eXtreme Programming
- xUnit, 73

- yellow bin, 369
- Yourdon, Edward, 54

- Z, 18, 256, 258
- zero-day attack, 203
- zero-defect, 377