
Index

Numbers

80-20 principle, 128. *See also* Pareto diagrams
99.999% availability, 360–361

A

Abstract class, OO technology, 332
Acceptance testing (vendor-developed software), 302–304
Accuracy. *See* Reliability
Active backlogs, 108
Active Design Reviews method, 180
Activity-based costs, process improvement, 462–464
Activity levels, process improvement, 462–466
Addition, interval scale and, 60
Alignment principle
 measuring value of process improvement, 447–448
 overview of, 443–444
Alternative-form method, reliability testing, 75
Alternative hypotheses, 483
American National Standards Institute (ANSI), 86
Analysis-driven data collection, 472
Analysis phase, OO model, 29
Analysis, TQM and, 8
ANSI. *See* American National Standards Institute
APAR. *See* Authorized program analysis report

Applicability, reliability models, 219
Appraisal Requirements for CMMI (ARC), 439
AS/400, 141–143, 193–194
Assessment. *See also* Malcolm
 Baldrige assessment; Quality assessment; Software project assessment
 vs. audit, 414–415
 internal assessments, 421–422
 ISO standard for, 414
 organizational-level, 416
 process improvement and, 454
 qualitative vs. quantitative, 455–456
 report, 429–433
 summary, products good enough to ship, 306–307
Attenuation, 76–77
Audits
 vs. assessments, 414–415
 ISO 9000, 49
 quality audits, 397
Authorized program analysis report (APAR), 139–140
Availability metrics, 359–374
 99.999% availability, 360–361
 availability compared with reliability, 363
 defect rates and, 364–366
 key factors in, 360

- Availability metrics (*continued*)
 - outage cause analysis, 369–370
 - outage data collection, 369, 371–372
 - outage indicators, 366
 - outage tracking forms, 367–368
 - outages in-process, 372–373
 - productivity and, 362
 - references, 373–374
 - reliability and, 362–364
 - summary, 373
 - system availability technologies, 363–364
- Average method size, 334–335
- B**
- Back end of development process, 56–57
 - defect tracking model for, 269
 - quality metrics and, 57
 - reliability models and, 256
- Backlog management index (BMI)
 - control charts and, 151
 - software maintenance and, 106–107
- Backlogs, defect. *See* PTR arrival and backlog model
- Baselines
 - process capability, 450
 - process improvement and, 455–456
- Basic measures
 - percentages, 63–65
 - proportions, 62–63
 - rates, 65–66
 - ratios, 62
 - six sigma, 66–70
- Benchmarks, 455–456. *See also* Baselines
- Beta program outages, 372–373
- Black box approach, software testing, 311
- Blitz testing, 255
- BMI. *See* Backlog management index
- BOOTSTRAP method, 416
- Bugs, software, 4, 86
- Business applications, availability, 361
- C**
- Calendar-time data, 210, 215, 252
- Capability
 - management models and, 257
 - reliability models and, 219
- Capability Maturity Model (CMM), 39–44
 - applications of, 40–41
 - CMM-based methods, 417–418
 - control charts and, 149
 - defect rates and, 365
 - defect removal effectiveness (DRE) and, 181–183
 - industry leadership and, 458–459
 - maturity questionnaire, 424
 - process improvement and, 437–438
 - process maturity levels, 39–40
 - six step cycle in, 417
 - Software Engineering Institute (SEI) and, 9
- Capability Maturity Model Integration (CMMI)
 - model based approach to process improvement, 438
 - process maturity levels, 42–44, 438
 - staged vs. continuous representations, 42, 44, 440–441
- Card and Glass model, 321–322
- CASE tools, 238, 484
- Causality, 79–82
 - correlation and, 79
 - criteria for measuring, 80–81
- Cause-and-effect diagrams
 - defined, 130
 - fishbone diagram example (Grady and Caswell), 152–153
- CBA IPI. *See* CMM-Based Appraisal for Internal Process Improvement
- CBO. *See* Coupling Between Object Classes
- CDF. *See* Cumulative distribution function
- Certification, ISO standard, 414
- Changed source instructions (CSI), 91
- Check sheets. *See* Checklists
- Checklists, 130–133
 - defined, 128
 - effectiveness of, 130–131
 - illustration of, 129
 - inspection checklist, 238
 - process compliance and, 449
 - types of, 131–133
- CHECKPOINT tool, SPR, 44
- CK (Chidamber and Kemerer) metrics suite, 337–343
 - applying to two companies, 338–339
 - combining with other metrics, 342
 - list of metrics in, 337–338
 - validation studies, 339–343
- Class hierarchy, OO technology, 332–334
- Classes, OO technology, 332, 347
- Classification, scientific, 59
- Cleanroom methodology, 32–35
 - development team control and, 32–34
 - front-end defect removal and, 180

- illustration of, 33
 - overview of, 32
 - stages of, 34–35
 - statistical testing and, 34
- Cleanroom Software Engineering, 32
- Cluster sampling, 379
- CMM. *See* Capability Maturity Model
- CMM-Based Appraisal for Internal Process Improvement (CBA IPI), 417–418, 438
- CMMI. *See* Capability Maturity Model Integration
- Code development phase, 242
- Code inspections. *See* Design review/code inspection (DR/CI)
- Code integration pattern, 242–245
 - early defect removal and, 245
 - integration over time, 243
 - middle of development process and, 269
 - project-to-project comparisons, 244
 - uses of, 244
- Code reuse. *See* Reuse
- Commitment
 - management commitment, 474
 - team commitment, 297
- Companies, customer satisfaction with, 388–389
- Complexity metrics. *See also* Structural complexity metrics
 - CK metrics suite, 336–337
 - compared with reliability and quality management models, 311
 - cyclomatic complexity, 315–318
 - Halstead’s software science, 314–315
 - lines of code (LOC) metric, 312–313
 - module design examples, 322–328
 - references, 329
 - structural metrics, 319–322
 - summary, 328
 - syntactic constructs for, 318–319
 - system model, 320–322
- Component quality analysis, 135–136, 369–370
- Component test (CT) phase, PTR backlogs, 295
- Components
 - availability, 362
 - defined, 193
 - testing, 17
- Compression factor, reliability models, 224–226
- Computer science vs. software science, 314
- Concepts, measurement theory, 58
- Concrete class, OO technology, 332
- Confidence intervals
 - reliability and, 203
 - software development and, 482
- Confidence level, survey samples and, 380–381
- “Conformance to requirements,” 2
- Construct validity, 71
- Content validity, 71–72
- Continuous representation (systems engineering), CMMI, 42, 44, 440–441
- Control charts, 143–152
 - defect removal effectiveness, 151–152
 - defects, 146–148
 - defined, 130
 - effective use of, 149–151
 - illustration of, 129
 - inspection effectiveness, 148–149
 - process improvements and, 149
 - as quantitative analysis tool, 402
 - SPC and, 143–144, 481–482
 - types of, 145–146
 - uses of, 145
- Control flow loops, 315–316, 318–319
- Correlation, 77–80
 - causality and, 79
 - logical guidance for use of, 81
 - metrics and, 76, 78–79
 - module defect level with other metrics, 324–325
 - outliers and, 80
 - weak relationships and, 78
- Cost effectiveness, defect removal effectiveness (DRE) and, 177–180
- Cost performance index (CPI), 152
- Coupling Between Object Classes (CBO), 338
- CPI. *See* Cost performance index
- CPU utilization, 286–289, 301
- Criteria for model evaluation, 207, 218–219, 257
- Criterion-related validity, 71
- Critical problems, 124, 293, 401
- Crosby, P. B., 2, 7, 9
- CSI. *See* Changed source instructions
- Cultural aspects, TQM, 8
- Cumulative distribution function (CDF), 189, 208, 214
- CUPRIMDA, 381–382
- CUPRIMDSO, 4, 98
- Curvilinear model, size to defect rate, 312–313

- Customer-Centered Six Sigma* (Naumann and Hoisington), 69–70
- Customer problems metric, 96–99
 - problems per user month (PUM), 96–97
 - relationship of defects and customer satisfaction, 97–98, 99
- Customer programs. *See* Early customer programs (ECP)
- Customer satisfaction, 375–395
 - companies and, 388–389
 - customer problems metric and, 97–98
 - data analysis, 381
 - data collection, 376–377
 - defects and customer problems and, 99
 - histogram analysis, 137
 - how much is enough, 390–392
 - market share and, 390–391
 - metrics, 98–100
 - quality and, 4–5
 - references, 393–394
 - sample size, 379–381
 - sampling methods, 377–379
 - small organizations and, 392–393
 - specific areas vs. overall satisfaction, 382–388
 - summary, 393
 - surveys, 98, 371, 376–381
 - UPRIMD categories, 383–388
- Customers
 - cost of recruiting vs. cost of retaining, 375
 - feedback from, 376–377
 - as focus of TQM programs, 8
 - perspective on defect rates, 92–93
 - quality evaluation by, 3, 255
- Cyclomatic complexity (McCabe), 315–318
 - correlation with LOC, 317
 - formula for, 315
 - program testability and maintainability, 315–316
 - scatter diagrams, 140–141
 - uses of, 317–318
- D**
- Data
 - balancing analysis and collection, 472
 - complexity, 320–321
 - gathering and testing, 57–58
 - quality control and, 470–472
 - software engineering. *See* Software engineering data
 - summarization format, 407
- Data collection
 - customer satisfaction surveys and, 376–377
 - data quality and, 470
 - methodology (Basili and Weiss), 118
 - questionnaire, 423
- Data quality, 203, 470–472, 475
- Defect arrival data, 223–224. *See also* PTR
 - arrival and backlog model
 - distributing overtime, 226–229
 - exponential model and, 209
 - OO projects and, 349–350
 - outcome indicators and, 299
 - reliability growth models and, 254
- Defect arrival rate
 - machine testing and, 101–102
 - software testing and, 279–282
- Defect arrivals curve, 282
- Defect attributes, orthogonal defect classification (ODC), 268
- Defect backlog. *See* PTR arrival and backlog model
- Defect data, 119–123
 - documentation defects, 122–123
 - inspection defects, 119
 - interface defects, 119
 - logic defects, 121
- Defect density
 - customer problems metric and, 97–98
 - customer satisfaction and, 99
 - lines of code (LOC) and, 312
 - product quality metric and, 86–87
- Defect discovery periods, orthogonal defect classification (ODC), 267
- Defect injection
 - activities, 164–165
 - by phases, 165–171
- Defect origin, 159, 166, 262–263
- Defect origin-where found matrix, 169–171
- Defect prevention. *See also* Defect prevention process
 - CMM model and, 41
 - Rayleigh model and, 236
- Defect prevention process (DPP), 35–39
 - applying to waterfall model, 38
 - IBM's success with, 36–37
 - illustration of, 37
 - key elements of, 35–36
 - steps in, 35, 131
- Defect rates. *See also* Six sigma
 - availability metrics and, 364–366
 - defective parts per million (DPPM), 69

- machine testing and, 100–101
 - mean time to failure (MTTF) and, 364
 - OO quality and, 347–348
 - by phase, 265–266
 - process improvement and, 461
 - software and, 66
- Defect removal
- activities, 164–165
 - code integration pattern and, 245
 - model evaluation and, 257
 - patterns, 239–240
 - phase-based, 103, 165–171
 - Rayleigh model and, 236
- Defect removal effectiveness (DRE), 159–185
- control charts and, 151–152
 - cost effectiveness and, 177–180
 - defect injection and removal activities, 164–165
 - defect injection and removal by phases, 165–171
 - formulas for measuring, 160–161
 - literature review, 160–164
 - overview of, 103–105, 159–160
 - phase-based defect removal model, 172–174
 - process maturity levels and, 181–183
 - references, 184–185
 - summary, 183–184
 - two-phase model, 174–177
- Defect removal model (DRM)
- cost effectiveness of, 177–180
 - defect removal patterns and, 239–240
 - examples of, 173
 - overview of, 172–174
 - two-phase model, 174–177
- Defect triggers, orthogonal defect classification (ODC), 267–268
- Defective fixes, 110
- Defective parts per million (DPPM), 66–67
- Defects
- backlog over time, 283–284
 - by category (Pareto analysis), 133–134
 - by component (Pareto analysis), 135
 - control charts and, 146–148
 - defined, 86
 - distribution of, 295–296
 - exponential model and, 209
 - ODC classification, 267
 - by OO class, 347
 - relationship to failure, 87
 - by severity (histogram analysis), 136
 - by testing phases, 297
 - tracking and reporting, 258
- Defects per thousand sources lines of code (KLOC), 66
- Definitions, 58
- Delayed S model, 215–216
- Delinquent fixes, 138–139
- Department of Defense (DoD), 365, 415
- Deployment, 448
- Depth of Inheritance Tree (DIT), 338
- Design
- high-level design (HLD), 16, 258
 - low-level design (LLD), 16, 258
- Design phase, OO model, 28
- Design review/code inspection (DR/CI), 238, 255
- Developers, data provided by, 475
- Development. *See also* Object-oriented development
- environment for OO projects, 352–353
 - iterative model. *See* Iterative development process
 - measurement and, 484
 - methodologies, 413
 - practices for OO projects and, 355
 - Rayleigh model and, 480–481
 - SPC and, 481–483
- Development cycle
- early indicators, 400
 - quality assessment and, 399
- Development phases
- integration phase, 56
 - multiple models for evaluating, 257
 - quality indicators by, 401
- Discrepancy reports (DR), 161–162
- Distribution of defects, 295–296
- DIT. *See* Depth of Inheritance Tree
- “Do it right the first time” principle, 192, 236
- DO WHILE statements, 319
- Documentation
- customer satisfaction and, 384
 - defects, 122–123
 - process improvement and, 446–447
- DoD. *See* Department of Defense
- Downtime
- availability metrics and, 360
 - MTTR and, 363
- DPP. *See* Defect prevention process (DPP)
- DPPM. *See* Defective parts per million
- DR/CI (Design review/code inspection), 238, 255
- DR. *See* Discrepancy reports

- DRE. *See* Defect removal effectiveness
 DRM. *See* Defect removal model
 Dynamic reliability models, 187–188
- E**
- Early customer programs (ECP), 18–19, 274–275, 305
 Early defect removal
 code integration pattern and, 245
 model evaluation and, 257
 phase-based defect removal, 103
 Rayleigh model and, 236
 Early detection percentage, 161–162
 ECP. *See* Early customer programs
 Education, OO projects, 351–352
 Effort/defect matrix, 242, 259, 261
 Effort indicators
 difficulty of establishing, 301
 effort/outcome model and, 298
 S curve and, 299
 Effort/outcome model, 298–302
 effort indicators and outcome indicators, 298
 example scenarios, 299–300
 improvement actions, 301
 matrix structure of, 299
 quality management and, 241, 402
 test coverage and scoring, 301–302
 Engineering. *See* Software engineering
 Entry-Task Validation-Exit (ETVX) paradigm, waterfall model, 14
 Error detection efficiency. *See* Defect removal effectiveness
 Errors
 defined, 86
 detection rate, 208
 injection rate, 239–240
 listing, 131
 measurement errors, 73–74
 ETVX. *See* Entry-Task Validation-Exit paradigm, waterfall model
 European Foundation for Quality Management, 47
 European Quality Award, 47
 Evaluation phase, quality assessment, 402–405
 evaluation criteria, 405
 qualitative data, 403–405
 quantitative data, 402–403
 Evolutionary prototyping, 21
 Execution-time data, 210
 Executive leadership, 8
- Expert opinions, 403, 405
 Exponential model, 208–211
 as case of Weibull family, 208
 ease of implementation, 209
 predictive validity, 211
 reliability and survival studies and, 208
 software reliability and, 208–209
 testing and, 209–210
 Extreme Programming (XP), 31–32
 Extreme-value distributions, 189
- F**
- Face-to-face interactions, 428
 Face-to-face interviews, 376
 Fact gathering
 phase 1—project assessment, 422–423
 phase 2—project assessment, 425–426
 Fagan’s inspection model, 180
 Failures
 defined, 86
 factors in project failure, 426–427
 instantaneous failure rate, 208
 relationship to defects, 87
 Fan-in/fan-out metrics (Yourdon and Constantine), 319–320
 Fault containment. *See* Defect removal effectiveness
 Fault count models
 assumptions, 218
 reliability growth models and, 212
 Faults, 86
 Field defects
 arrival data, 228–229
 compared with tested defects, 236–237
 Fishbone diagrams, 130, 152–153. *See also* Cause-and-effect diagrams
 “Fitness for use,” 2
 “Five 9’s” availability, 360–361
 Fix backlog metric, 106–107
 Fix quality, 109–110
 Fix response time metric, 107–108
 Flow control loops, 316–317, 319–320
 FPs (Function points), 56. *See also* Function point metrics
 Frequency bars. *See* Pareto diagrams
 Frequency of outages, 360
 Front end of development process, 56–57, 269
 Function point metrics, 93–96
 background and overview of, 456
 compared with LOC counts, 93–94

- example defect rates, 96
- issues with, 95
- measuring process improvements at
 - activity levels, 462–466
 - OO metrics and, 334
 - opportunities for error (OFE) and, 93
- Function points. *See* FP
- Functional defects, 267
- Functional verification test (FVT), 422–423
- Functions, 93
- FURPS (Hewlett-Packard customer satisfaction), 4, 98
- G**
- Goal/Questions/Metric paradigm (Basili and Weiss), 110, 472
- Goel generalized nonhomogeneous Poisson process model, 214
- Goel-Okumoto (GO) Imperfect Debugging model, 213, 217
- Goel-Okumoto Nonhomogeneous Poisson Process (NHPP) model, 213–215
- Goodness-of-fit test (Kolmogorov-Smirnov), 222
- GQM. *See* Goal/Questions/Metric paradigm (Basili and Weiss)
- Graphic methods, modeling and, 253
- H**
- Halstead's software science, 314–315
 - formulas for, 314–315
 - impact on software measurement, 314–315
 - primitive measures of, 314
- Hangs
 - effort/outcome model and, 301
 - software testing and, 289–291
- Hazard rate, 208
- Hewlett-Packard
 - defects by category, 134
 - fix responsiveness, 108
 - FURPS customer satisfaction, 98
 - software metrics programs, 115–116
 - TQC program, 7
- High availability. *See also* Availability metrics
 - defect rates and, 365
 - importance of, 359
 - reliability and, 362
- High-level design (HLD), 16, 258
- Histograms
 - defined, 129
 - examples of, 136–138
 - illustration of, 129
 - as quantitative analysis tools, 402
- HLD. *See* High-level design
- Human cultural aspects, TQM, 8
- Hypotheses
 - formulating and testing, 57
 - statistical methods and, 483
- I**
- I Know It When I See It* (Guaspari), 3
- IBM
 - eServer iSeries, 273
 - Market Driven Quality, 8
 - Object Oriented Technology Council (OOTC), 336
 - OS/2 and IDP, 26–27
 - programming architecture, 14
 - Sydney Olympics (2000) project, 302–303
- IBM Federal Systems Division, 191, 201
- IBM Houston, 161
- IBM Owego, 25
- IBM Rochester
 - OO projects of, 336–337
 - software metrics programs, 116–117, 273–276
- IBM Rochester software testing, 273–276
 - assigning scores to test cases, 275
 - defect tracking, 279
 - process overview, 273
 - reliability metric, 291
 - showstopper parameter (critical problems), 293
 - system stress tests, 286–287
 - testing phases, 274
- IDP. *See* Iterative development process
- IE. *See* Inspection effectiveness
- IF-THEN-ELSE statements, 316, 318–319
- IFPUG. *See* International Function Point Users Group
- Implementation phase, OO model, 28
- Improvement actions. *See also* Process improvement
 - module design and, 326–328
 - by project, 432–433
 - project assessment and, 427–428
 - software testing and, 295–297
- In-process availability metrics, 372–373
- In-process quality assessment. *See* Quality assessment

- In-process quality metrics, 100–105
 - defect arrivals, 101–102
 - defect density, 100–101
 - defect rate by phase, 265–266
 - defect removal effectiveness, 103–105
 - effort/outcome model and, 241
 - inspection reports, 258–263
 - inspection scoring questionnaire, 261
 - overview of, 100
 - phase-based defect removal, 103
 - Rayleigh model and, 237, 258–260
 - unit test coverage and defect report, 264–265
- In-process software testing metrics, 271–309
 - acceptance testing (vendor-developed software), 302–304
 - CPU utilization, 286–289
 - defect arrivals, 279–282
 - defect backlog, 283–284
 - effort/outcome model and, 298–302
 - implementation recommendations, 294–295
 - improvement actions and, 295–297
 - mean time to unplanned IPLs, 291–293
 - product preparedness for shipping, 304–307
 - product size over time, 285
 - quality management and, 297–298
 - references, 309
 - showstopper parameter (critical problems), 293
 - summary, 308–309
 - system crashes and hangs, 289–291
 - test progress S curve, 272–279
- Index of variation, 71
- Industry leadership factors, 458–459
- Inflection S models, 215–216
- Information flow metric, 320–321
- Infrastructure, process improvement stages, 457–458
- Inheritance, OO technology, 332
- Inheritance tree depth, 334
- Initial program loads (IPLs), 289–293
- Injection rate, errors, 239–240
- Inspection checklist, 238
- Inspection defects, 119, 317
- Inspection effectiveness (IE), 148–149, 167, 240–241
- Inspection effort, 240–242, 258–262, 269, 481
- Inspection model (Fagan), 180
- Inspection reports, 258–263
 - defect origin and type, 263
 - effort and defect rates, 260
 - effort/defect matrix, 259
- Inspection scoring checklist, 269
- Inspection scoring questionnaire, 261–262
- Instance variables, OO technology, 332
- Instantaneous failure rate, 208
- Instruction statements. *See* Lines of code (LOC)
- Integrated product and process development (IPPD), 42
- Integration phase, development process, 56
- Interface defects, 119, 262, 312
- Intermodule complexity, 321–322
- Internal assessments, 421–422
- Internal consistency method, 75
- International Function Point Users Group (IFPUG), 95, 456, 462
- International Organization for Standardization (ISO), 47, 414
- International Software Quality Exchange (ISQE), 471
- Interval scale, 60–61
- Interviews, qualitative data and, 400, 403
- Intramodule complexity, 321–322
- IPL tracking tools, 290
- IPLs. *See* Initial program loads
- IPPD. *See* Integrated product and process development
- Ishikawa's seven basic tools. *See* Quality tools
- ISO 9000, 47–51
 - audits, 49
 - changes to, 51
 - compared with Malcolm Baldrige Assessment, 50
 - document control requirements of, 48
 - elements of, 47–48
 - software metrics requirements, 49
- ISO. *See* International Organization for Standardization
- ISQE. *See* International Software Quality Exchange
- Iterative development process (IDP), 24–27
 - IBM OS/2 development as example of, 26–27
 - illustration of, 25
 - overview of, 24–25
 - steps in, 26
- Iterative enhancement (IE). *See* Iterative development process
- Iterative model, 24–27

J

- Jelinski-Moranda (J-M) model, 212–213, 216–217
- Joint application design (JAD), 457
- Juran, J. M., 2, 5

K

- Key process areas (KPA), 40
- KLOC (defects per thousand sources lines of code), 66
- Kolmogorov-Smirnov goodness-of-fit test, 222

L

- Lack of Cohesion on Methods (LCOM), 338
- Languages. *See* Programming languages
- Leadership
 - project teams and, 428
 - TQM and, 8
- Lean Enterprise Management, 10
- Least-squares
 - correlation, 78
 - PTR arrival and backlog model and, 252
- Library of generalized components (LGC), 30
- Library of potentially reusable components (LPRC), 30
- Life cycle, software, 191
- Life of product (LOP), 88
- Likert scale, 56–57
- Lines of code (LOC)
 - complexity metrics and, 312
 - correlation with cyclomatic complexity (McCabe), 316–318
 - defects relative to, 88–92
 - defined, 88
 - example defect rates, 91–92
 - function points and, 93–94, 456
 - measurement theory and, 56
 - OO metrics and, 334
 - tracking over time, 285
 - variations in the use of, 88–89
- Literature review
 - defect removal effectiveness (DRE), 160–164
 - project assessment, 426–427
- Littlewood (LW) models, 213, 217
- Littlewood nonhomogeneous Poisson process (LNHPP) model, 213
- LLD. *See* Low-level design
- LOC. *See* Lines of code

- Logic defects, 121
- Lonenz metrics and rules of thumb, 334–336
- Loops, 316–319
- LOP. *See* Life of product
- Low-level design (LLD), 16, 258
- LPRC. *See* Library of potentially reusable components
- LW. *See* Littlewood models

M

- M-O (Musa-Okumoto) logarithmic Poisson execution time model, 215
- Machine testing, 100–101
- Mailed questionnaire, 376, 394
- Malcolm Baldrige assessment, 45–47
 - assessment categories, 45
 - award criteria, 46
 - ISO 9000 and, 49–50
 - purpose of, 46
 - used by US and European companies, 46–47
 - value of feedback from, 46
- Malcolm Baldrige National Quality Award (MBNQA), 7, 45
- Management, projects. *See* Project management
- Management, quality. *See* Quality management models; Total quality management
- Management technologies, 456–457
- Managerial variables, CK metrics, 340–341
- Manpower buildup index (MBI), 200
- Margin of error, sample size, 379–380
- Mark II function point, 95, 462
- Market Driven Quality, IBM, 8
- Market share, customer satisfaction and, 390–391
- Mathematical operations
 - interval scale, 60
 - ratio scale, 61
- Maturity. *See also* Capability Maturity Model (CMM); Capability Maturity Model Integration (CMMI); Process maturity project assessment and, 416 software evaluation based on, 415
- MBI. *See* Manpower buildup index
- MBNQA. *See* Malcolm Baldrige National Quality Award
- McCabe's complexity index. *See* Cyclomatic complexity
- Mean time between failure (MTBF). *See* Mean time to failure (MTTF)

- Mean time to failure (MTTF)
 - cleanroom methodology and, 34
 - defect rates and, 364
 - as quality metric, 86–87
 - as reliability metric, 291, 362
- Mean time to repair (MTTR), 363, 366
- Mean time to unplanned IPLs (MTI), 291–293
 - example of, 292–293
 - formula for, 291–292
 - used by IBM Rochester as reliability measure, 291
- Means, 323
- Measurement. *See also* Metrics
 - data quality control and, 470–472
 - errors, 73–74
 - future of, 484
 - references, 485
 - software quality and, 472–475
 - SPC and, 481–483
 - state of art in, 484
 - TQM and, 8
- Measurement theory, 55–83
 - attenuation, 76–77
 - causality, 80–82
 - concepts and definitions and, 58
 - correlation, 77–80
 - errors, 73–74
 - example application of, 56–58
 - interval and ratio scales, 60–61
 - nominal scale, 59
 - operational definitions, 58–59
 - ordinal scale, 59
 - percentages, 63–65
 - proportions, 62–63
 - rates, 65–66
 - ratios, 62
 - references, 83
 - reliability, 70–73, 75–76
 - scientific method and, 55–56
 - six sigma and, 66–70
 - summary, 82–83
 - validity, 71–72
- Messages, OO technology, 332
- Methods, OO technology, 332
- Metrics. *See also* Measurement
 - availability. *See* Availability metrics
 - back end quality and, 57
 - Capability Maturity Model (CMM), 41
 - CK. *See* CK metrics suite
 - complexity. *See* Complexity metrics
 - correlation and, 76
 - function points. *See* Function point metrics
 - module design. *See* Module design metrics
 - OO projects and, 348
 - in process quality. *See* In-process quality metrics
 - product quality. *See* Product quality metrics
 - productivity. *See* Productivity metrics
 - software maintenance. *See* Software maintenance metrics
 - software programs and, 472–475
 - software quality. *See* Software quality metrics
 - software testing. *See* In-process software testing metrics
 - structural complexity. *See* Structural complexity metrics
 - team commitment and, 297
- Metrics programs
 - Hewlett-Packard, 115–116
 - IBM Rochester, 116–117
 - Motorola, 110–115
- Middle of development process, code integration pattern, 269
- Models, 475–481. *See also* by individual type
 - criteria for evaluating, 479
 - deficiencies in application of, 477–478
 - development process and, 480
 - empirical validity and, 479
 - list of, 475–476
 - neural network computing technology and, 478–479
 - probability assumptions and, 478
 - professional's use of, 476–477
 - steps in modeling process, 220–224
 - types of, 476
- Module design metrics, 322–328. *See also* Complexity metrics
 - correlation between defect level and other metrics, 323–324
 - defect levels, 324–325
 - defect rate, 325–326
 - identifying improvement actions, 326–327
 - list of metrics used, 323
 - means and standard deviations of variables, 324
 - measuring defect level among program modules, 323–324
- Modules. *See* Program modules

- Motorola
 - Quality Policy for Software Development (QPSD), 110–114
 - Six Sigma Strategy, 7–8, 66–67
 - Software Engineering Process Group (SEPG), 110–114
 - software metrics programs, 110–115
- MTBF (mean time between failure). *See* Mean time to failure
- MTI. *See* Mean time to unplanned IPLs
- MTTF. *See* Mean time to failure
- MTTR. *See* Mean time to repair
- Multi-inspector phase, 180
- Multiple regression model, 326–327
- Multivariate methods, 402
- Musa-Okumoto (M-O) logarithmic Poisson execution time model, 215
- N**
- NASA, Software Assurance Technology Center (SATC), 342
- NASA Software Engineering Laboratory, 118
- NEC Switching Systems Division, 103
- Net satisfaction index (NSI), 99–100
- Neural network computing technology, 478–479
- NOC. *See* Number of Children of a Class
- Nominal scale, 59
- Nonhomogeneous Poisson Process (NHPP) model, 213–215
- Nonlinear regression, 195
- NSI. *See* Net satisfaction index
- Null hypotheses, 483
- Number of Children of a Class (NOC), 338
- O**
- Object-oriented analysis (OOA), 331
- Object-oriented design (OOD), 331
- Object-oriented (OO) development, 27–32
 - Branson and Hernes methodology for, 26–29
 - design and code reuse and, 29–30
 - Extreme Programming (XP) and, 31–32
 - overview of, 27
 - phases of, 27–29
 - steps in, 28–29
 - Unified Software Development Process and, 30–31
- Object-oriented programming (OOP), 331
- Object-oriented projects, 331–358
 - CK metrics, 337–343
 - education and skills required, 351–352
 - examples of, 336–337
 - Lonenz metrics and rules of thumb, 334–336
 - metrics applicable to, 348
 - OO concepts and constructs, 331–334
 - OO metrics, 342–343
 - performance and, 355
 - productivity and, 343–347
 - project management for, 353–354
 - quality and development practices and, 355
 - quality management, 347–351
 - references, 357
 - reusable classes and, 354–355
 - summary, 356–357
 - tools and development environment for, 352–353
- Object Oriented Technology Council (OOTC), 336
- Objects, OO technology, 332
- ODC. *See* Orthogonal defect classification
- OFE. *See* Opportunities for error
- OO development. *See* Object-oriented development
- OOA. *See* Object-oriented analysis
- OOD. *See* Object-oriented design
- OOP. *See* Object-oriented programming
- OOTC. *See* Object Oriented Technology Council
- Operational definitions, 58–59
- Opportunities for error (OFE), 87, 93
- Ordinal scale, 59
- Organizational-level
 - assessment, 416
 - management commitment at, 474
- Organizations, 416
- Orthogonal defect classification (ODC), 266–269
 - based on defect cause analysis, 266
 - defect attributes, 268
 - defect discovery periods, 267
 - defect triggers, 267–268
 - defect types and, 267
- OS/2, 26
- Outage duration. *See* Mean time to repair
- Outages
 - analyzing causes, 369–370
 - data collection methods, 369, 371–372
 - duration of, 360
 - frequency of, 360
 - incidents and downtime, 370

- Outages (*continued*)
 - indicators derived from outage data, 366
 - tracking forms for, 367–368
- Outcome indicators. *See also* Effort/outcome model
 - commonly available, 301
 - defect arrival patterns and, 299
 - effort/outcome model and, 298
- Outliers
 - correlation and, 80
 - PTR arrival and backlog model, 250–251
- Overall satisfaction, 382–388
- P**
- p* charts, 145
- Pareto diagrams, 133–136
 - defects by component, 136
 - defined, 128
 - examples of, 134–135
 - illustration of, 129
 - quantitative analysis tools, 402
 - return on investment and, 133
- PCE. *See* Phase containment effectiveness
- PDCE. *See* Phase defect containment effectiveness
- PDF. *See* Probability density function
- PDPC. *See* Process decision program chart
- Pearson correlation coefficient
 - assumption of linearity, 78
 - complexity metrics and, 323
 - Rayleigh model and, 194
- Peer reviews
 - process compliance and, 449
 - project assessment, 423
 - software development and, 414
- Percent delinquent fixes, 108–109, 138–139
- Percentages, 63–65
- performance, OO projects, 355
- Personal Software Process (PSP), 42
- Phase-based defect removal model (DRM).
 - See* Defect removal model
- Phase containment effectiveness (PCE), 163–164, 169
- Phase defect containment effectiveness (PDCE), 170
- Phase effectiveness, 104–405
- Plan-Do-Check-Act, 8–9
- Platform availability. *See* Availability metrics
- Poisson distribution, 145
- Polynomial model, 250, 476
- Precision. *See* Validity
- Predictive validity, 203, 218
- Preparation phase, project assessment, 421–422, 425–426
- Preparation phase, quality assessment
 - qualitative data, 400–402
 - quantitative data, 399–400
- Probability assumption, 478
- Probability density function (PDF), 189–190, 208
- Probability sampling methods, 377–379
 - cluster sampling, 379
 - simple random sampling, 377–378
 - stratified sampling, 378–379
 - systematic sampling, 377–378
- Problem tracking reports (PTRs), 221, 279–282. *See also* PTR arrival and backlog model; PTR submodel
- Problems per user month (PUM), 96–97
- Process adoption, 448–449
- Process assessment, 455–456
- Process capability
 - levels of, 440
 - measuring, 440
- Process compliance, 449–450
- Process decision program chart (PDPC), 154
- Process improvement, 453–467
 - activity levels and, 462–466
 - alignment principle, 443–444
 - CMMI staged vs. continuous representations, 440–441
 - day by day, 450–451
 - defect rates and, 461
 - economics of, 459–461
 - maturity assessment models, 438–440
 - measuring, 447–448
 - process adoption, 448–449
 - process capability levels, 440
 - process compliance, 449–450
 - process documentation, 446–447
 - process maturity levels, 438
 - productivity and, 461
 - references, 452, 467
 - schedules and, 461
 - six stage program, 454–459
 - summary, 451, 466–467
 - teams, 441–443
 - time to market goals and, 444–445
 - TQM and, 8
- Process maturity, 437–452. *See also* Capability Maturity Model (CMM) assessment models, 438–440

- compared with project assessment, 415–417
 - framework and quality standards, 39
 - levels of, 438
 - measuring, 438–440
 - Processes, process improvement stages and, 457
 - Product level
 - vs. module level, 311
 - reliability, 188
 - Product quality
 - reliability and defect rate as measure of, 364
 - software quality and, 3–4
 - vs. total quality management, 375
 - Product quality metrics, 86–100
 - customer problems metric, 96–98
 - customer satisfaction metric, 98–100
 - customer's perspective, 92–93
 - defect density metric, 87–88
 - defect density vs. mean time to failure, 86–87
 - function points and, 93–96
 - intrinsic product quality and customer satisfaction, 86
 - lines of code (LOC) metric, 88–92
 - Product size over time, 285
 - Productivity metrics, 343–347
 - availability metrics and, 362
 - examples, 343–345
 - limitations of, 345
 - person-days per class, 343–344
 - process improvement and, 461
 - two-dimensional vs. three- or four-dimensional measures, 346–347
 - units of measurement for, 343
 - Products, good enough to ship, 304–307
 - indicators or metrics for, 304–305
 - negative metrics and, 305
 - product type and, 304
 - quality assessment summary, 306–307
 - Program length formula, 82
 - Program modules. *See also* Module design metrics
 - complexity metrics and, 311
 - module design examples, 322–328
 - Program size, 312
 - Program temporary fix (PTF) checklist, 131–133
 - Programming languages
 - formal specification languages, 20
 - fourth generation languages, 21
 - UML (Unified Modeling Language), 30, 457
 - variations in LOC, 89
 - Programming Productivity* (Jones), 88–89
 - Project assessment. *See* Software project assessment
 - Project charters, 421–422
 - Project closeout plans, 422
 - Project level, vs. module level, 311
 - Project management
 - improvement actions and, 427
 - in-process quality assessment and, 399
 - OO projects, 353–354
 - Proportions, 62–63
 - Prototyping
 - OO development and, 354
 - risk-driven approach and, 23
 - Prototyping model, 19–21
 - evolutionary prototyping, 21
 - factors in quick turnaround of prototypes, 20–21
 - overview of, 19
 - rapid throwaway prototyping, 21
 - steps in, 20
 - Pseudo-control charts. *See* Control charts
 - PSP. *See* Personal Software Process
 - PTF (program temporary fix) checklist, 131–133
 - PTR arrival and backlog model, 249–253
 - applications of, 249–250
 - backlogs over time, 283–284
 - component test (CT) phase and, 295
 - contrasted with exponential model, 250
 - defect arrivals over time, 279–282
 - outliers and, 251–252
 - predictability of arrivals with, 252–253
 - predictor variables, 250
 - PTR submodel, 245–249
 - deriving model curve, 247–248
 - not applicable for projection, 248
 - testing defect tracking, 246
 - using in conjunction with other models, 249
 - variables of, 246
 - PTRs. *See* Problem tracking reports
 - PUM. *See* Problems per user month
- Q**
- QIP. *See* Quality improvement program
 - QPSD. *See* Quality Policy for Software Development

- Qualitative data, 400, 400–405, 402–403, 405, 447
- Quality. *See also* Software quality
 - ambiguity regarding, 1
 - of assumptions, 219
 - audits, 397
 - of conformance, 3
 - customer satisfaction as validation of, 375
 - of design, 3
 - expense and, 2
 - formula (conformance to customers' requirements), 4, 5
 - improvement strategies, 238
 - in-process assessment. *See* In-process quality assessment
 - metrics and, 297–298
 - OO projects and, 355
 - popular view of, 1–2
 - professional view of, 2–3
 - projections with Rayleigh model, 237
 - six sigma as measure of, 66–70
- Quality assessment, 397–411
 - assessment ratings over time, 408
 - evaluation phase, 402–405
 - overview of, 397–399
 - parameters of, 406
 - preparation phase, 399–402
 - products, good enough to ship, 306–307
 - recommendation phase, 408–410
 - references, 411
 - scale, 407
 - summarization phase, 406–408
 - summary, 410
- Quality Improvement Paradigm/Experience Factory Organization, 9
- Quality improvement program (QIP), 254–255
- Quality management models, 235–270
 - code integration pattern, 242–245
 - compared with complexity metrics, 311
 - evaluating, 257
 - in-process metrics and reports, 258–266
 - OO projects and, 347–351
 - orthogonal defect classification, 266–269
 - PTR arrival and backlog model, 249–253
 - PTR submodel, 245–249
 - Rayleigh model, 236–242
 - references, 270
 - reliability growth models, 254–257
 - summary, 270
- Quality metrics. *See* In-process quality metrics
- Quality of measurement
 - attenuation, 76–77
 - errors, 73–74
 - reliability, 70–73, 75–76
 - validity, 71–72
- Quality parameters, 5
- Quality Policy for Software Development (QPSD), 110–114
- Quality tools, 127–158
 - cause-and-effect diagrams, 152–153
 - checklists, 130–133
 - control charts, 143–152
 - histograms, 136–138
 - new, 154
 - overview of, 127–130
 - Pareto analysis, 133–136
 - references, 158
 - relations diagrams, 154–156
 - running charts, 138–140
 - scatter diagrams, 140–143
 - summary, 156–157
- Quantitative analysis tools, 402
- Quantitative data
 - quality assessment evaluation phase, 402–403
 - quality assessment preparation phase, 399–400
- Questionnaires
 - CMM maturity questionnaire, 424
 - customer feedback and, 376–377
 - project assessment and, 423–425, 487–508
- R**
- R charts, 145
- RAISE system tests, IBM Rochester, 286–288
- Random errors, 73, 75
- Random sampling, 377–378
- Rank-order correlation, 78
- Rapid throwaway prototyping, 21
- Rates, 65–66
- Ratio scale, 60–61
- Ratios, 62
- Rayleigh model, 189–206
 - assumptions, 192–195
 - development process and, 480–481
 - illustration of, 192
 - implementation of, 195–203
 - in-process quality metrics and, 258–260
 - references, 206
 - reliability and validity and, 203–205
 - software life cycle and, 191

- summary, 205
- underestimation and, 204
- Weibull distribution and, 189–190
- Rayleigh model, quality management, 236–242
 - defect prevention and early removal, 236
 - error injection rate and, 239–240
 - inspection process and, 240–242
 - organization size and, 269
 - quality improvement strategies and, 238
 - quality projections, 237
 - relationship between testing and field defect rates, 236–237
- Real-time delinquency index, 109
- Reboots. *See* Initial program loads
- Recommendation phase, quality assessment
 - recommendations, 408–409
 - risk mitigation and, 409–410
- Recommendations, software project assessment, 428–429
- References
 - availability metrics, 373–374
 - complexity metrics, 330
 - customer satisfaction, 393–394
 - defect removal effectiveness, 184–185
 - in-process quality assessment, 411
 - in-process software testing, 309
 - measurement, 485
 - measurement theory, 83
 - object-oriented projects, 357
 - process improvement, 452, 467
 - quality management, 270
 - quality tools, 158
 - Rayleigh model, 206
 - reliability growth models, 231–233
 - software development process models, 52–54
 - software project assessment, 435
 - software quality, 11–12
- Relations diagrams, 154–156
- Relative cost, 118, 183
- Reliability, 70–76. *See also* Rayleigh model
 - assessing, 75–76
 - availability and, 362–364
 - defined, 70–71
 - measurement errors and, 73–74
 - measurement quality and, 73
 - operational definition of, 362
 - random errors and, 75
 - technologies for improving, 363–364
 - validity and, 72, 203–205
- Reliability growth models, 207–233
 - assumptions, 216–218
 - compression factor, 224–226
 - defects over time, 226–229
 - evaluating, 218–219
 - exponential model and, 208–211
 - Goel-Okumoto Imperfect Debugging model, 213
 - Goel-Okumoto Nonhomogeneous Poisson Process (NHPP) model, 213–215
 - Jelinski-Moranda (J-M) model, 212–213
 - Littlewood models, 213
 - modeling process, 220–224
 - Musa-Okumoto logarithmic Poisson execution time model, 215
 - overview of, 207, 211–212
 - references, 231–233
 - S models, 215–216
 - summary, 229–231
- Reliability growth models, quality management, 254–257
 - advantages and applications of, 256
 - defect arrival patterns, 254
 - quality improvement program, 254–255
- Reliability models. *See also* Rayleigh model
 - compared with complexity metrics, 311
 - deficiencies in application of, 477–478
 - for small organizations, 230–231
 - static and dynamic, 187–188
- Removal efficiency, 96, 160–161, 402, 443, 465
- Remus-Zilles, 174, 177–178
- Reports, assessment, 429–433
- Reports, in-process
 - defect rate by phase, 265–266
 - inspection, 258–263
 - inspection scoring questionnaire and, 261
 - severity distribution of test defects, 266
 - test defect origin, 266
 - unit test defects, 264–265
 - unit test defects by phase, 265–266
- Response for a Class (RFC), 338
- Response time, software maintenance, 107–108
- Results, software project assessment, 428–429
- Return on investment (ROI), 133
- Reuse
 - analyzing with scatter diagrams, 141–143
 - design and code, 29–30
 - list of reusable artifacts, 458
 - OO projects and, 354–355
 - reusable software parts, 20
- RFC. *See* Response for a Class

- Rigorous implementation, 56
- Risk analysis, spiral development model and, 23
- Risk exposure rates, 65
- Risk mitigation, 409–410
- Rules of thumb (Lorenz), 334–336
- Run charts, 138–140
 - defined, 130
 - illustration of, 129
 - percent of delinquent fixes, 138–139
 - tracking cumulative parameters, 140
 - uses of, 138
- S**
- S curve, 140, 272–279, 299
- S models, 145, 215–216
- Sample size, customer surveys, 379–381
 - absolute size vs. relative size, 380–381
 - margin of error and, 379–380
- Sampling methods, 377–379
 - cluster sampling, 379
 - simple random sampling, 377–378
 - stratified sampling, 378–379
 - Systematic sampling, 377–378
- SATC. *See* Software Assurance Technology Center
- Satisfaction scale, 381, 383. *See also* Customer satisfaction
- Scales of measurement
 - hierarchical nature of, 61
 - interval, 60–61
 - nominal, 59
 - ordinal, 59
- SCAMPI. *See* Standard CMMI Appraisal Method for Process Improvement
- Scatter diagrams, 140–143
 - analyzing nonlinear relationships, 78
 - defined, 129–130
 - illustration of, 129
 - McCabe's complexity index and, 140–141, 324–325
 - as quantitative analysis tools, 402
 - relationship of reuse to defects, 141–143
 - uses of, 140
- SCE. *See* Software Capability Evaluation
- Schedule performance index (SPI), 152
- Scheduled uptime, 360
- Schedules, process improvement and, 461
- Scientific method, 55–56
- Scope creep, 285
- Scope of project assessment, 413, 415–416
- SEI. *See* Software Engineering Institute
- SELECT statements, 319
- SEPG. *See* Software Engineering Process Group
- Server availability, 360–361
- Service calls, outage data and, 369
- SETT. *See* Software Error Tracking Tool
- Seven basic quality tools. *See* Quality tools
- Severity distribution of test defects report, 266
- Shipped source instructions (SSI), 91
- Showstopper parameter (critical problems), 293
- Significance tests, 483
- Simple random sampling, 377–378
- Simplicity, reliability models and, 219
- Single-inspector phase, 180
- Six sigma, 66–70
 - centered vs. shifted, 69
 - implications for process improvement and variation reduction, 69–70
 - as industry standard, 66
 - Motorola, 7–8
 - standard deviation and, 66–67
- Six stage program, process improvement overview, 454
 - Stage 0: Software Process Assessment and Baseline, 455–456
 - Stage 1: Focus on Management Technologies, 456–457
 - Stage 2: Focus on Software Processes and the Methodologies, 457
 - Stage 3: Focus on New Tools and Approaches, 457
 - Stage 4: Focus on Infrastructure and Specialization, 457–458
 - Stage 5: Focus on Reusability, 458
 - Stage 6: Focus on Industry Leadership, 458–459
- Skills
 - metrics expertise, 474–475
 - OO projects, 351–352
 - skill-building incentives, 446–447
- SLIM. *See* Software Life-cycle Model tool
- Small organizations
 - customer satisfaction and, 392–393
 - defect removal effectiveness and, 182–183
 - object-oriented metrics and, 356
 - quality management for, 269
 - reliability modeling for, 230–231
 - software testing recommendations, 308
- Small teams
 - complexity metrics and, 327–328
 - OO projects and, 356

- Software
 - complexity. *See* Complexity metrics
 - control charts and, 149
 - development models. *See* Software development process models
 - development process models
 - getting started with metrics program for, 472–475
 - life cycle, 191
 - maintenance metrics. *See* Software maintenance metrics
 - measurement. *See* Measurement
 - process improvement. *See* Process maturity
 - productivity. *See* Productivity metrics
 - project assessment. *See* Software project assessment
 - quality management. *See* Quality management models
 - quality metrics. *See* Software quality metrics; Software quality metrics
 - quality models. *See* Models
 - reliability. *See* Reliability
 - stress tests, 286–287
 - testing metrics. *See* In-process software testing metrics
- Software Assessments, Benchmarks, and Best Practices* (Jones), 96
- Software Assurance Technology Center (SATC), 342
- Software Capability Evaluation (SCE), 438
- Software development process models, 13–54
 - cleanroom methodology, 32–35
 - defect prevention process, 35–39
 - ISO 9000, 47–51
 - iterative model, 24–27
 - Malcolm Baldrige assessment, 45–47
 - object-oriented approach, 27–32
 - process maturity framework and quality standards, 39
 - prototyping approach, 19–21
 - references, 52–54
 - SEI maturity model, 39–44
 - spiral model, 21–24
 - SPR assessment method, 44–45
 - summary, 51–52
 - waterfall model, 14–19
- Software engineering
 - CMMI model, 440
 - software development and, 470
 - software quality and, 475–481
 - techniques and tools, 484
- Software engineering data, 117–123
 - challenge of collecting, 117
 - collection methodology, 118
 - defect data and, 119–123
 - expense of collecting, 118
- Software Engineering Institute (SEI)
 - assessment vs. capability evaluations, 415
 - CBA IPI, 417–418
 - CMM, 9, 39–44
 - DPP and, 39
 - lack of solid data on process improvements, 453
 - rate of software flaws, 365
 - SCAMPI assessors, 439
- Software Engineering Metrics and Models* (Conte), 88
- Software Engineering Process Group (SEPG)
 - alignment principle, 443–444
 - capability baseline, 450
 - measuring process improvement, 447–448
 - monitoring process adoption, 448–449
 - Motorola, 114–115
 - not focusing only on maturity level, 442–443
 - as process improvement team, 441
 - reliable goals and, 445
- Software Error Estimation Reporter (STEER), 201–203
- Software Error Tracking Tool (SETT), 222
- Software Life-cycle Model (SLIM) tool, 200
- Software maintenance metrics, 105–110
 - fix backlog and backlog management index, 106–107
 - fix quality, 109–110
 - fix response time and responsiveness, 107–108
 - overview of, 105
 - percent delinquent fixes, 108–109
- Software Productivity Research, Inc. (SPR), 44–45, 454. *See also* SPR assessment method
- Software project assessment, 413–436
 - activities by phases of, 433–434
 - assessment report, 429–433
 - audits vs. assessments, 414–415
 - characteristics of approach to, 420–421
 - CMM-based method, 417–418
 - facts gathering phase 1, 422–423
 - facts gathering phase 2, 425–426
 - improvements for, 426–428
 - preparation phase, 421–422

- Software project assessment (*continued*)
 - project assessment vs. process maturity assessment, 415–417
 - questionnaire for, 423–425
 - references, 435
 - results and recommendations, 428–429
 - scope of, 413
 - SPR method, 419–420
 - summary, 434–435
 - summary report, 433
- Software quality, 1–12
 - customer's role in, 3–4
 - overview of, 4–7
 - popular view of, 1–2
 - professional view of, 2–3
 - references, 11–12
 - summary, 10
 - TQM and, 7–10
- Software Quality and Productivity Analysis (SQPA), 7
- Software Quality Assurance (SQA) group, 449
- Software quality engineering, 475–481
- Software quality metrics
 - data collection, 117–123
 - Hewlett-Packard program, 115–116
 - IBM Rochester program, 116–117
 - Motorola program, 110–114
 - overview of, 85–86
 - in process quality. *See* In-process quality metrics
 - product quality. *See* Product quality metrics
 - references, 125–126
 - software maintenance. *See* Software maintenance metrics
 - summary, 123–125
- Space Shuttle software project, 161, 194–195, 238
- SPC. *See* Statistical process control
- Spearman's rank-order correlation, 78, 194
- SPI. *See* Schedule performance index
- Spiral development model, 21–24
 - advantages/disadvantages of, 23–24
 - illustration of, 22
 - risk analysis and risk-driven basis of, 23
 - TRW Software Productivity System, 21
- Split-halves method, 75
- SPR assessment method, 44–45
 - assessment themes, 45
 - five-point scale of, 44
 - process maturity and, 416
 - questionnaire topics, 44–45
 - steps in, 419–420
- SPR. *See* Software Productivity Research, Inc.
- Spurious relationships, causality, 81
- SQA. *See* Software Quality Assurance group
- SQC. *See* Statistical quality control
- SQPA. *See* Software Quality and Productivity Analysis
- SSI. *See* Shipped source instructions
- Staged representation (software engineering), CMMI, 42, 44, 440–441
- Standard CMMI Appraisal Method for Process Improvement (SCAMPI), 415, 418, 438–439
- Standard deviation
 - module design metrics and, 324
 - sample size and, 380–381
 - six sigma measures and, 66–70
- Standards, ISO 9000, 47–51
- Static measures vs. dynamic, 65
- Static reliability models, 187–188
- Statistical methods
 - null hypotheses and alternative hypotheses and, 483
 - software development and, 482
- Statistical process control (SPC), 143–144, 481–483
- Statistical quality control (SQC), 481–483
- STEER. *See* Software Error Estimation Reporter
- Stratified sampling, 378–379
- Stress testing, 286, 301
- Structural complexity metrics, 319–322
 - defined (Henry and Kafura), 319–320
 - fan-in/fan-out metrics (Yourdon and Constantine), 319–320
 - interaction between modules and, 320
 - system complexity model (Card and Glass), 320–322
- Subtraction, interval scale and, 60
- Success of projects, factors in, 426–427
- Summarization phase, quality assessment
 - overall assessment, 406–408
 - strategy for, 406
- Summary report, project assessment, 433
- Supplier quality, 390
- Surveys, customer satisfaction, 98, 371, 376–381
- Survival studies, exponential model and, 208
- SVT. *See* System verification test

- Sydney Olympics (2000), acceptance testing and, 302
 - Syntactic constructs, complexity metrics, 318–319
 - System availability. *See* Availability metrics
 - System complexity model (Card and Glass), 320–322
 - System crashes and hangs
 - effort/outcome model and, 301
 - software testing and, 289–291
 - System tests
 - CPU utilization and, 286–289
 - quality improvement program and, 255
 - System verification test (SVT), 422–423
 - Systematic errors, 73, 75
 - Systematic sampling, 377–378
 - Systems engineering, CMMI model, 440
- T**
- TDCE. *See* Total defect containment effectiveness
 - Team commitment, project success and, 297
 - Team Software Process (TSP), 42
 - Teams
 - complexity metrics and, 327–328
 - process improvement and, 441–443
 - project assessment and, 427–429
 - quality management models for, 474
 - Telephone interviews, 376
 - Test case, metrics, 303–304
 - Test defect origin report, 266
 - Test execution, metrics, 303–304
 - Test plan curve, 278–279
 - Test progress S curve
 - assigning scores to test cases and, 275
 - coverage weighting, 276
 - execution plan and, 278
 - illustration of, 275
 - plan curve and, 278–279
 - purpose of, 272
 - tracking, 276–278
 - Test/retest method, 75, 76
 - Tests
 - assigning scores to test cases, 275
 - component, 17
 - defect removal and, 160
 - defect tracking and, 246, 269
 - determining end date for, 256
 - effort/outcome model and, 298
 - functional verification test (FVT), 422–423
 - Kolmogorov-Smirnov goodness-of-fit test, 222
 - phases, 295
 - for significance, 483
 - system verification test (SVT), 422–423
 - tested defects vs. field defects, 236–237
 - unit tests (UT), 16–17, 243, 264–266
 - Time between failures models
 - assumptions, 217–218
 - Jelinski-Moranda (J-M) model, 212–213
 - reliability growth models and, 212
 - reliability models and, 197
 - Time boxing, 21
 - Time to market goals, 444–445
 - Tools. *See also* Quality tools
 - OO projects, 352–353
 - process improvement stages and, 457
 - Total defect containment effectiveness (TDCE), 163
 - Total Quality Control (TQC), 7
 - Total quality management (TQM)
 - background of, 7
 - customer satisfaction and, 375
 - Hewlett-Packard's program, 7
 - IBM's program, 8
 - key elements of, 8, 392
 - Motorola's program, 7–8
 - organizational frameworks for, 8–10
 - TQC. *See* Total Quality Control
 - TQM. *See* Total quality management
 - Tracking defects. *See also* problem tracking reports (PTRs)
 - IBM Rochester software testing, 279
 - OO quality and, 347–348
 - PTR submodel and, 246
 - testing and, 269
 - Tracking outage data, 367–368
 - Training materials, 447
 - Trend charts, 402
 - Trillum model, 416
 - TRW Software Productivity System (TRW-SPS), 21–22
 - TSP. *See* Team Software Process
- U**
- u* charts, 145–146
 - Unified Modeling Language (UML), 30, 457
 - Unified Software Development Process, 30–31
 - Unit tests (UT)
 - code development and, 243
 - coverage and defect report, 264–265

- Unit tests (UT) (*continued*)
 - defects by test phase report, 265–266
 - waterfall model, 16–17
- Unplanned IPLs. *See* Initial program loads
- UPRIMD (usability, performance, reliability, installability, maintainability, documentation, and availability) categories, 383–388
- Use-case model, UML, 30
- UT. *See* Unit tests

- V**
- Validation of data, 118
- Validity, 70–73
 - defined, 71
 - measurement quality and, 73
 - predictive vs. empirical, 203–204
 - reliability and, 72
 - systematic errors and, 75
 - types of, 71–72
- Variation, index of, 71
- Vendors. *See also* Acceptance testing (vendor-developed software)
- Video conferencing, 428
- Vienna Development Method (VDM), 34

- W**
- Waterfall development model, 14–19
 - advantages of, 14
 - coding stage, 16
 - component tests, 17
 - early customer programs (ECP), 18–19
 - Entry-Task-Validation-Exit (ETVX), 14
 - example implementation, 14–15
 - high-level design, 16
 - low-level design, 16
 - system-level test, 17–18
 - unit tests, 16–17
- Weibull models
 - exponential model, 208–211
 - Rayleigh model, 189–190
- Weighted Methods per Class (WMC), 337–338
- Where found, 169–171, 182–183
- Workload characteristics, 286–287

- X**
- X-bar charts, 145
- XP (Extreme Programming), 31–32

- Z**
- Z notation, 34
- Zero point, ratio scale and, 61

InformITTM

www.informit.com

- Articles
- Books
- Free Library
- Expert Q&A
- Training
- News
- Downloads

OPERATING SYSTEMS
 WEB DEVELOPMENT
 PROGRAMMING
 NETWORKING
 CERTIFICATION
 AND MORE...

Solutions from experts you know and trust.

**Expert Access.
 Free Content.**

- Free, indepth articles and supplements
- Master the skills you need, when you need them
- Choose from industry leading books, ebooks, and training products
- Achieve industry certification and advance your career
- Get answers when you need them from live experts or InformIT's comprehensive library



Visit **InformIT**
 and get great content

from 
**Addison
 Wesley**

Addison-Wesley and InformIT are trademarks of Pearson plc / Copyright©2000 pearson

www.informit.com



Register Your Book

at www.awprofessional.com/register

You may be eligible to receive:

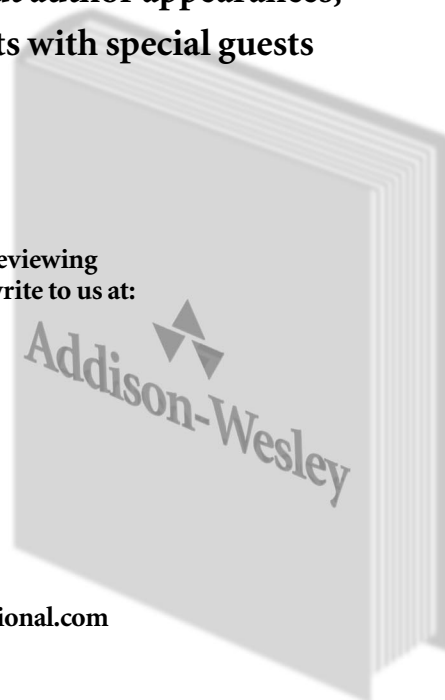
- Advance notice of forthcoming editions of the book
- Related book recommendations
- Chapter excerpts and supplements of forthcoming titles
- Information about special contests and promotions throughout the year
- Notices and reminders about author appearances, tradeshows, and online chats with special guests



Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com



Visit us on the Web: <http://www.awprofessional.com>