

Index

- 3" × 5" card, 110, 199
5" × 7" card, 199, 349
- Abstraction 262, 328, 356
Act out the software 112
Act small and early 83
Active critical inquiry 135
Active voice 22
Actual cost 72–73, 75
Adams, Scott 49
Agile Manifesto 173
Agile methods 8, 52, 164, 173–174,
277–278, 313, 393–395
Agile modeling 277
Analysis
 context 111
 detailed 333
 feasibility 333
 global 334
 impact 29, 32, 43, 449, 452–453
 requirements 98–99, 108, 111, 466
 risk 239–240, 250–251, 384–386, 388
 structured 265, 274, 459–460, 466
Annual pizza lunch 51
Arbitrary plans 158
Architecture 265
Assumptions 137
Attributes 106–107
Audit, functional 30, 450, physical 30,
450
Auditing 28–29, 440, 450–452
Automation boundary 466
- Backlog of features 176
Baseline
 allocated 27, 32, 164, 281–282, 316,
325, 338, 357, 391, 442–444,
447, 449, 451
 design 27, 29, 32, 42–43, 90, 164,
281, 325, 338, 360, 391,
442–443, 445, 449, 451
 development 27, 29, 32, 42, 44, 325,
338, 360–361, 442–443, 445,
451
 functional 26, 29, 32, 42–43, 89–90,
123, 140, 143–144, 164,
226–227, 280, 282, 316, 338,
357, 371, 378, 382–383, 387,
391, 441–445, 449, 451
 operational 27, 32, 43–44, 164, 338,
391, 442–443, 446, 451
 product 27, 32, 43–44, 164, 338, 361,
391, 442–443, 445, 451
 requirements 280, 282
Baselines 26, 28–31, 33, 89, 143–144,
164, 166, 193, 280–282, 314,
328, 338, 379, 385, 393,
440–441, 455
Bebugging 294
Beck, Kent 50, 174, 313
Behavioral model 461, 463–464
Best practices 7, 9, 180, 187–189, 230
Blame 49
Boehm, Barry 168, 239, 242, 383
Brainstorming 24, 112, 118

- Brooks, Frederick 261
Business event 102
Business perspective 6
Buzan, Tony 119, 472
- Capability Maturity Model 4, 8, 14, 109,
159, 180, 182, 184–189,
191–193, 228, 230, 247, 255,
339, 459, 472
Capability Maturity Model Integrated 8,
11, 180, 187
- Card
3" × 5" 110, 199
5" × 7" 199, 349
CRC 19, 112, 124, 130, 274
index 174
story 174, 176
task 199
Cards-on-the-wall 19, 207–208,
222–223, 372, 383
Cause-effect graphing 303, 310
CCB *see* Configuration Control Board
CCPM *see* Critical Chain Project
Management
Change request 56
Charette, Robert 246
CI *see* Configuration Item
CM sketch 33–34, 440, 455
CM staff 29–30, 32–33, 55, 63, 75,
80–81, 84, 90, 144, 227, 233,
282–284, 286, 314, 318,
325–326, 336, 338, 353, 357,
359–361, 371, 378–379, 382,
384, 441, 449–450, 452–453,
456
CMM *see* Capability Maturity Model
CMMI *see* Capability Maturity Model
Integrated
Coad, Peter 459
Cockburn, Alistair 164
Cohesion 266
Co-locate 10
Commercial Off the Shelf 278–280, 377,
379, 382–385
Commitments 156, 222
Compassion 5
Computer Software Component 446,
448
Computer Software Configuration Item
446
Computer Software Unit 446, 448
Concept of Operations 111, 118–119,
135, 354–355, 371, 446
Concurrent creation 118
Configuration Control Board 25, 29,
31–32, 42–43, 75, 84, 98, 104,
140, 144, 193, 227, 254–255,
282, 316–317, 325, 338, 356,
358, 360–361, 372, 382, 391,
395, 441, 449, 451–453
Configuration Item 445–448
ConOps *see* Concept of Operations
Constraints 106–107, operational 466
Context analysis 111
Context diagram 19, 111, 116, 124, 196,
274, 461–462
Control 28, 52, 440, 448
Control events 461
Control gate 352–353, 355–361,
371–372, 374–375, 379,
382–383
Control gate question 352
Cost Performance Index 72–73
Cost variance 73
COTS *see* Commercial Off the Shelf
Coupling 265, 268, 356
Cox, Jim 86
CPI *see* Cost Performance Index
CPM *see* Critical Path Management
CRC cards 19, 112, 124, 130, 274
Create, Retrieve, Update, and Delete
112, 268, 349
Crisis management 246
Critical chain 205
Critical Chain Project Management
204–207
Critical path 203–204, 207
Critical Path Management 203–204
Crosby, Philip 86
CRUD *see* Create, Retrieve, Update, and
Delete
CSC *see* Computer Software Component
CSCI *see* Computer Software
Configuration Item
CSU *see* Computer Software Unit
Cumulative project cost 65
Curtis, Bill 186

- Daily builds 168, 308
- Daily SCRUM meeting 176
- Daily smoke test 167
- Data abstraction 266–268, 275
- Data dictionary 124, 461, 463–464
- Data store 462–464, 467–469
- Dataflow diagram 19, 124, 196, 265, 274, 462–464, 467, 469
- Davis, Alan 97
- DBWA *see* Design by walking around
- Decision book 347
- Decision table 268, 272
- Decomposition description 286
- DeMarco, Tom 265, 473
- Dependency description 286
- Derby, Esther 240
- Design by contract 276
- Design by walking around 115–117, 473
- Design CCB 281
- Design during requirements 108
- Design in the code 275–276
- Design recovery 325
- Design
 - box-structured 310
 - functional 213
 - detailed 287, 357, 442–443
 - high-level 281–282, 285–287, 315–316, 356–357, 360, 372, 384, 387, 390–392, 443–444, 447, 451
 - low-level 281–282, 285–287, 315, 338, 356–357, 359–360, 384, 386, 387, 390–392, 451
 - preliminary 158, 356, 372
 - structured 466–467
- Detailed analysis 333
- Deterioration dynamics 339
- DFD *see* Dataflow diagram
- Diagram
 - context 19, 111, 116, 124, 196, 274, 461–462
 - dataflow 19, 124, 196, 265, 274, 462–464, 467, 469
 - Entity-Relationship 464
 - environment 460
 - Hierarchical Input Process Output 124, 127–128, 274–275, 334, 356
 - sequence 131, 274
 - state transition 19, 129–130, 274, 302, 464
 - Warnier-Orr 19, 124, 128–129, 274–275
- Diagram of effects 121–122
- Difference that makes a difference 54
- Discrepancy Report 331
- Disguising the smell 54
- Do what you said you what do 101
- Document 19, 21, 41, 44, 134
- Document what you touch 336–337
- DoD-Std 2167A 228–229
- DR *see* Discrepancy Report
- Draw me a picture 51
- DWYSYWD *see* Do what you said you what do
- Earned value 72–73, 75
- Earned value tracking 69–72, 75, 88
- Easter eggs 298
- Edge of time 273
- Emery, Dale 195
- Emic interviewing 101
- Empowerment 49
- Encapsulation 266
- Entity-Relationship Diagram 464
- Environment 48, 76–78, 81, 86, 348, 375
- Environment diagram 460
- Environmental model 460
- Equivalence partitioning 303
- ERD *see* Entity-Relationship Diagram
- Error guessing 302
- Error seeding 294
- Error-free software 293, 296
- Error-prone module 296
- Essential model 460, 465–467
- Estimating games 158
- Estimation 156–157, 191, 208, 220
- Event list 461
- Evolutionary delivery 15
- Evolutionary process 40, 141, 161, 164, 166, 170, 223, 230, 277, 362, 371, 374–375, 377, 395
- EVT *see* Earned value tracking
- Executive sponsor 26, 33, 104, 113, 190, 196, 353, 355–356, 360–361, 371–374, 378–379, 382–384, 387–388, 446, 453

- Expectations 49, 105–106, 156
Extreme programming 174, 176, 186,
277–278, 313, 395
- Facilitated meetings 112–113
Facilitator 82, 114–115, 117–118
Failure 50–51
Fanaticism 51
FAST 113
Fate 51
Fatuousness 50
Fatware 107
Feasibility analysis 333
Final Operational Capability 210
Fit criteria 133
Fix-on-failure 246
Flow-oriented events 461
FOC *see* Final Operational Capability
Folly 50
Forward engineering 324
Frailty 50
Fraud 51
Freedman, Daniel 22
Fun 51
Functional CCB 281
Functional specification 111
Functions 106
- Gantt chart 63, 356
Gemmer, Art 238
Gilb chart 120, 137, 268, 271–272
Gilb, Tom 55, 120, 299, 472
Glass, Robert 97
Global analysis 334
Goal, question, metric 59
Going dark 54
Good enough software 167
Gottesdiener, Ellen 113
Grady, Robert 58
Graham, Dorothy 55, 299, 472
Group memory 21
- Heathkit 320
Heavy methodologies 172
Hierarchical Input Process Output
diagram 124, 127–128, 274–275,
334, 356
HIPO diagram *see* Hierarchical Input
Process Output
- Humphrey, Watts 58, 180, 182, 297,
471
- ICD *see* Interface Control Document
ICWG *see* Interface Control Working
Group
IDD *see* Interface Design Document
IDEA *see* Inspection, Demonstration,
Execution, and Analysis
Identification 28, 440, 448
IEEE 12207 23, 35, 230
IEEE Std 1042–1987 26, 35, 453, 472
IEEE Std 1058 231–232, 354
IEEE Std 1219 331, 334, 338, 230
IEEE Std 829 318, 320
IEEE Std 830–1998 35, 98, 145, 148
IEEE-Std-1016 282, 284, 286
Ignorant originality 11
-ilities 137
Impact analysis 29, 32, 43, 449,
452–453
Inch-pebble 90
Incremental consensus 394
Independent Verification and Validation
299
Index card 174
Information hiding 266, 268, 274, 328,
356
Initial Operational Capability 210
Inspection 55, 193
Inspection, Demonstration, Execution,
and Analysis 99, 192, 292–293,
296–299, 317–318, 390
Integration and test(ing) 291–322
big bang 305 307
bottom-up 305–307
build 305, 307
sandwich 305–306
top-down 305–307
Interface Control Document 446–447
Interface Control Working Group 447
Interface description 286
Interface Design Document 447
Interface Requirements Specification
447
Interviewing 100–101, 103
IOC *see* Initial Operational Capability
ISO 9000 14
Isorisk chart 251

- Iteration 161, 261, 383, 445
Iterative process 164, 166, 170, 230,
277, 395
IV&V *see* Independent Verification and
Validation
- Jackson, Michael 134
JAD *see* Joint Application Development
Joint Application Development
113–115, 165, 466
Jones, Capers 58
Journal 347
- Kaner, Sam 113
Kerth, Norman 82
Key process area 185
- Lean software development 174, 177,
179
Learning cycle 170
Light methodologies 172
Linear regression 218
Lister, Timothy 473
Logging meeting 56
Long-life software 328
- Maguire, Steven 86
Maintenance
adaptive 324–325, 327, 330, 373–
374
corrective 324–325
emergency 324, 327
perfective 324–325, 327, 330
Maintenance policy document 340
Management Information Center 60,
62–63, 81, 84, 110, 160, 192,
233, 296, 348, 350, 356–357,
360–361, 371–372, 378, 382,
391–393, 445
Management secrets 174
Marciniak, John J. 471
McCarthy, Jim 168
McConnell, Steve 17, 266, 327, 472
Meetings 5, 103
Mellor, Stephen 459
Member commitment 21
Memorandum 348, 353, 361, 371, 378
Mentoring 168
Metrics 8, 157, 209, 240
Metrics etiquette 58, 180
MIC *see* Management Information
Center
Microsoft process 166–167, 324
Milestone 89–90, 192, 350
Miller, Robin 392
Mil-Std-498 23, 35, 230
Mindful bureaucracy 8
Mindless bureaucracy 6
Mindmap 19, 21, 119–120, 472
Misunderstanding 20
Model
behavioral 461, 463–464
environmental 460
essential 460, 465–467
processor 467, 469
program implementation 469
Rayleigh 209, 211, 213, 215,
223–224, 473
Satir Change 194
systems implementation 467
task 468
User implementation 466
Modeling 142
Modification Request 331, 333–334,
336, 338–340, 449
Modularity 266–267, 274, 328
MR *see* Modification Request
Multitasking 207
Myers, Ware 179, 209, 244–245, 473
- Natural language modeling 142
Network of tasks 63, 199, 207–208, 222,
233, 356, 372, 383
- O&M *see* Operations and Maintenance
O'Bryan, Roy 474
Objects, functions, and states 112,
273–274
Operations and Maintenance 141, 279
Optimistic 209
- Pair programming 174–175, 195
Paper prototype 165
Parnas, David 259
PASE *see* Post-Its Assisted Software
Engineering
Passive-aggressive 105
People skills 5

- Perpetual pregnancy 242
Personal qualities 5
Personal Software Process 8, 11, 58, 157,
180–182, 209, 215–216, 219,
297, 313, 356
PERT Chart 198, 356
Pessimistic 209
Phillips Law of Software Half Life 17,
326
Pilot project 394
Plan-driven methods 52
Planned methods 173
Planned value 72–73, 75
Planning game 174, 176
Pleasant and Productive 51
Post-Its Assisted Software Engineering
23
Postmortem 184, 361
Pressman, Robert 22
Prime directive 82
Probe 215–216, 218–220
Process 11
Process improvement 179, 193, 195,
215, 348
Process specification 464
Processor model 467, 469
Product specification 167
Productivity number 245
Productivity parameter 211–212,
214–215
Program implementation model 469
Project context document 111, 119,
378
Project plan 69
Project web page 63
Prototype(ing) 105, 112, 164–165, 243,
285
 evolutionary 165
 throw-away 165
 rapid 112, 123, 164, 383–384, 445
Proxy-based Estimation *see* Probe
PSP *see* Personal Software Process
Public project progress poster 192
Putnam, Larry 179, 209, 244–245, 473

Quality assurance 293
Quality control 192
Quality, internal 16, 17 external 16
Quantitative chart 19

RAD *see* Rapid Application Development
Rapid Application Development
165–166
Rayleigh curve 66, 210, 213
Rayleigh model 209, 211, 213, 215,
223–224, 473
Reading and Writing 51
Recorder 23
Redocumentation 325
Reengineering 325
Refactoring 175, 277, 313, 325
Request for change 32, 42–43, 331, 449
Request for Proposal 135
Requirements analysis 98–99, 108, 111,
466
Requirements documentation 108
Requirements engineer 98–100,
105–106, 110–111
Requirements engineering 134
Requirements gathering 98–99, 108,
111, 123
Requirements management 98, 108
Requirements manager 110
Requirements tracking 75
Requirements triage 107
Requirements Verification Traceability
Matrix 133, 140, 280, 283–284,
314, 317–318, 334, 390, 395,
449–450
Requirements
 explicit 136
 external 197
 derived 136
 functional 106–107
 non-functional 106–107, 164
 project 197
 structured 467
 system 197
 unstated 136
 what-if 136
Resistance as a resource 195
Restructuring 325
Retrospective 81–82, 348
Return on Investment 179
Return on time invested 104
Reuse 267–268, 274
Reverse engineering 325, 334
Review 55, 193
Review board 110

- Review speed 56
RFC *see* Request for Change
RFP *see* Request for Proposal
Risk analysis 239–240, 250–251,
384–386, 388
Risk assessment 160
Risk aversion 246–247, 249–250, 255
Risk character, extent, and timing 249
Risk control 246
Risk directing 248
Risk estimation 248
Risk evaluation 248
Risk identification 240, 248
Risk leverage 251
Risk management 96, 192, 237–256
Risk manager 246, 248, 255
Risk mitigation 247
Risk planning 246
Risk referent 249, 251
Risk, sources of 57
Robertson, James 99
Robertson, Suzanne 99
ROI *see* Return on Investment
ROTI *see* Return on time invested
Rush, Gary 113
RVTM *see* Requirements Verification
Traceability Matrix
- SA/SD *see* Structured Analysis/Structured
Design
Sabotage 86
SADT *see* Software Analysis and Design
Technique
Satir Change Model 194
Scenario planning 253
Scenarios 142
Schedule Performance Index 72
Schedule variance 73
Schenck, Peter 100
SCRUM 174, 176–177, 194, 395
SCRUM Master 176–177, 194
SEI *see* Software Engineering Institute
Self-esteem 4, 11, 54
Sequence diagram 131, 274
SHAPE Forum 7, 474
Shared vision 21
Short sentences 22
Size status chart 63
Slack 203–204
Slip-lead chart 76–77
Smith, Dave 60
Soft status 53
Software Analysis and Design Technique
19, 124–127
Software decay 339
Software Design Description 282, 284,
286–287, 329, 356–358, 372,
383, 395, 448
Software Design Document 273, 336
Software design specification 184
Software Engineering Institute 4, 8, 109,
159, 185, 187, 247, 459, 472
Software equation 211, 245
Software Productivity Consortium 168,
170–171
Software Program Manager's Network
188
Software Requirements Specification 90,
97, 133–141, 143–146, 264, 273,
280, 284, 336, 355–357, 360,
371–372, 375, 382, 395, 446,
448, 465
SPC *see* Software Productivity
Consortium
SPI *see* Schedule Performance Index
Spiral process 15, 84, 168–169, 172,
243, 376, 379, 382–384
Spiral project 225–226
SPMN *see* Software Program Manager's
Network
Sprint 176–177
SRS *see* Software Requirements
Specification
SST *see* System Storyboarding
Technique
Staffing status 66
Standard deviation 221
Standish Group 6
State transition diagram 19, 129–130,
274, 302, 464
Statement of purpose 461
Status 52
Status accounting 28, 30–31, 55, 90,
440, 451–452
Status report 32, 452
Sticky note 10, 23–24, 117–118, 127,
199
Story card 174, 176

- Storyboard 19
- String 199
- Structure chart 19, 124–125, 274, 285, 334, 469
- Structured Analysis 265, 274, 459–460, 466
- Structured design 267
- Strunk and White 22
- Student Syndrome 207
- Study group 87
- Subcontractor 88
- Success breeds success 50
- Surprise factor 250
- Synchronize and stabilize 168
- Synergy 23
- System design concept 197
- System Storyboarding Technique 99, 112, 117–118, 473
- Systems implementation model 467

- Task card 199
- Task model 468
- Task network *see* network of tasks
- TDD *see* Test-Driven Development
- Team Software Process 180, 182, 184
- Temporal events 461
- Test plan 318–319, 333, 355, 357, 371, 382, 440
- Test procedure 319–320, 333, 355, 357, 360, 371, 382
- Test readiness review 336
- Test report 359–360
- Test summary report 320
- Test(ing)
 - acceptance 300, 336
 - black-box 287 300–305, 310, 333
 - boundary values 303
 - branch 304
 - cleanroom 300, 309–312
 - clear box 304
 - condition 304
 - decision 304
 - finite state machine 303
 - integration 300
 - qualification 309
 - qual 309
 - random 302
 - regression 300, 309
 - repeatable 300
 - risk-based 302
 - structure 309
 - test readiness 302
 - user 336
 - user profile 302, 310
 - white-box 287, 300, 304–305, 333
- Test-driven development 313–314
- That's not it 106
- Theory W 171
- Time box(ing) 242–244, 354, 372, 382–383, 473
- Top-risks list 62
- Touchy-feely 98
- TQM 179
- Tracing 29, 32, 43, 449, 452–453
- Trawling for requirements 99
- Trust 51
- TSP *see* Team Software Process
- Tufte, Edward, 21

- UDF *see* Unit Development Folder
- Unit Development Folder 78, 80–81, 358–359
- Use case 102
- Use case scenario 302
- User implementation model 466
- User manual 354–356, 360–361, 371
- User's Manual 191, 200

- V&V *see* Verification and Validation
- Valid Question 53
- Validation 299, 308
- Validity condition 53
- Values 49
- V-chart 95–96, 161–163, 165–166, 182, 225, 305, 314, 316–317, 331, 349, 354–355, 357, 359–360, 362, 371–378, 384, 389
- VDD *see* Version Description Document
- Verbatims 101–102
- Verifiable 265
- Verification 299
- Verification and Validation 299

- Version Description Document 329, 448
- Vision statement 167
- Volere 99, 102, 106–107, 132
- Volere requirements shell 132–133

- Walk the wall 21
Walkthrough 55, 193
Walls 19, 21, 24, 63, 117, 208
Ward, Paul 459
Warm-up exercise 208
Warm-up exercise 24
Warnier-Orr diagram 19, 124, 128–129,
274–275
Waterfall 13, 15, 39–40, 141, 161, 165,
170, 182, 185, 223, 277, 349,
358, 362, 371–373, 375, 383,
385, 390, 392–393, 440, 444
WBS *see* Work Breakdown Structure
Weigers, Karl 141
Weinberg, Gerald 4, 14, 22, 35, 48, 105,
192, 331, 394, 473–474
Weiss, Edmund, 23
White board 24, 84, 208
White board risk management 254
Who, What, Where, When, Why, How,
and How Much 158–159
Win-win spiral 172
Work Breakdown Structure 199,
223–225, 356, 372
WWWWHH *see* Who, What, Where,
When, Why, How, and How
Much
XP *see* extreme programming
Y2K Problem 266, 326
Yarn 199, 208
Yourdon, Ed 265, 459, 474
Zahniser, Richard 21, 115, 243, 473
Zultner, Richard 101

