Index

A
acceptance
comparison with validation and verification, 328–333
defined, 513
as key systems engineering concept, 60–61
in qualifications chain, 329, 330
as term in testing and qualification, 327–328
acceptance plans, 147, 161, 162, 184, 294, 513
acceptance testing, 12, 14, 19, 51, 61, 328, 330–332, 342, 347, 349–353
Ackoff, R. L., 448, 468
acyclic digraphs, 123, 418
Adam, J. A., 467
adaptibility, defined, 265
adjacency, in graphs, 118, 121, 124, 129, 137, 139
Agha, Y., 311
agility, defined, 264
Agrawala, A. K., 232, 295, 296, 318
air bag restraint system, case study in stakeholders’ requirements, 188–189
Akao, Y., 401
Alber, I. E., 468
Alexander, C., 73
Alford, M., 27, 384
Allen, R. H., 249
allocated architecture
analyzing functional activation and control structure, 294–297
approaches for solving allocation problem, 281–286
for automated soda machine, 367–369
conducting performance and risk analysis, 297–298
defined, 275, 513
defining allocation problem, 281
defining functional activation and control structure, 294–297
deriving requirements, 288–294
developing, 274–304
documenting, 298–299
finishing allocation problem, 286–288
functional allocation principles, 283
allocated architecture (Continued)
and human vs. machine responsibilities, 281–283
major development activities, 275–276
mapping functions to components, 279–288
obtaining approval, 298–299
overview, 26–27, 28, 48, 49, 274–279
relationship to IDEF0 model, 87
tracing non-input/output requirements, 288–294
Ambler, S. W., 22, 24
analogies, 254, 310
analytic models, 71, 72
Anderson, T., 232
antisymmetric relations, 119, 126
Apollo 13, operational concepts and not finding errors, 167–168, 482–483
Apollo 13, case study in stakeholders’ requirements, 189–192
application entities (AEs), 318
application layer, OSI reference model, 317, 318, 320
apportionment, 289–290, 330, 513
approximation, defined, 513
architectures. See also allocated architecture; functional architecture; physical architecture
behavioral model, 26–27
data model, 26–27
documenting, 298–299
interface architecture, 48, 49
message passing, 310–311
network, 312–313
overview, 26–29
process model, 26–27
shared memory, 311–312
system allocated architecture, 48, 49
Arciszewski, T., 255
Ariane 5 case study, 355–356
Arrow’s impossibility theorem, 428
asymmetry, as property of unary relations, 126
attainable (requirements attribute), 163, 164, 513
attribute listing, 253–254
automated soda machine
external systems diagram, 358–361
functional architecture, 365–367
fundamental objectives, 361
fundamental objectives with swing weights, 364
integration and qualification, 370–371
interface design, 369
objectives hierarchy, 361
operational concept, 357–358
overview, 357
physical and allocated architectures, 367–369
stakeholders’ requirements, 362–365
Ayton, P., 414

B
Baier, C., 346
Barney, G. C., 458
Barrett, B. E., 407
Barron, F. H., 406, 407
Bayes rule, 412–414, 416
Baylin, E. N., 214
behavior diagrams, 384–387. See also function flow block diagrams (FFBDs)
behavior models
behavior diagrams, 384–387
control flow diagrams, 392
defined, 375, 513
finite-state machines and state-transition diagrams, 387–389
overview, 384
Petri nets, 393, 394–395
statecharts, 389–392, 393, 394
Beizer, B., 340, 341, 342
Bell Labs or Bell Telephone Laboratories, 6–7, 16, 17, 476
Bender, E. A., 448
Bertalanffy, L. v., 446, 448, 468
Berube, M. S., 4, 10, 76
Bias, R. G., 180
Big-bang integration process, 339
binary relations, 109, 522
bipartite graphs, 122, 133–134, 514
Bimbaum, J., 272
black box testing, 348, 349, 355, 514
Blanchard, B. S., 20, 183, 298
block definition diagrams, 94, 95, 96, 97
block diagrams
aircraft control system example, 255, 256
defined, 69
internal, 94, 96
in physical architecture
development, 255, 256
SysML semantics, 94–96
SysML syntax, 94–96
Blum, B. L., 145
Boar, B. H., 22
Boardman, J., 452, 453, 469
Bock, C., 26, 94
Boehm, B. W., 20, 21, 31, 32, 473
Bohm, C., 91
bottom-up integration process, 332, 333,
337, 338
Boulton, Matthew, 217
Braasch, M. S., 301
brainstorming, 253, 254
brainwritinggame, 254–255
brainwriting pool, 255
Brandt, J. C., 44
Braun, R., 455
Bresnick, T. A., 160, 179
Brooks, C. G., 166
Brown, C. M. L., 181
Browning, T., 284
Buede, D. M., 101, 146, 160, 179, 188,
216, 398, 401, 402, 406, 407
bus architecture, 312, 313

C
cardinality, in graphs, 121
Carnahan, J. V., 160, 179
cartesian product of two sets, A B, 108, 514
category class, CORE, 63
causal loop diagrams, 452–453
CCITT (International Telephone and
Telegraph Consultative
Committee), 315 centralized
architecture, 260, 262, 514
CFDs (control flow diagrams), 392
Chambers, G. J., 147
Chapanis, A., 181
Chapman, W. L., 231
Charbonneau, S. M., 389, 390, 391, 392,
393, 394
Checkland, P., 19, 452, 453, 469
Choisser, R. W., 160
Chrissis, M. B., 485
Chu, W. W., 286
Chusho, T., 349
circuits, in digraphs, 122
CI s (configuration items), 5, 6, 10, 11, 12,
13, 14, 15
Clarke, L. A., 349
classes, defined, 23
Clausing, D., 264
client-server architecture
defined, 263, 514
as model for CORBA, 320
and morphological box, 251–252
closed-loop control processes, 217, 218
Cockburn, A., 168
common object request broker architecture
(CORBA) standards, 320–323
comparable (requirements attribute), 161,
163, 164, 514
complete (requirements attribute), 161, 163,
164, 187, 514
complete functionality, 206
component class, CORE, 63
components
defined, 514
deriving qualification requirements, 294
deriving trade-off requirements,
291–294
as key systems engineering concept, 57, 60
mapping functions to, 279–288
conceptual validity
defined, 74, 328–329, 514
in qualifications chain, 330, 331
role in validation, 331
concise (requirements attribute), 164, 514
concurent engineering, 8
configuration items (CI s), 5, 6, 10, 11, 12,
13, 14, 15, 514
connectedness, in graphs, 123–124
Connell, J. L., 179
consistent (requirements attribute), 161,
163, 164, 185, 514
constraint theory, 464–467, 514
context of a system. See system context
control flow diagrams (CFDs), 392
control systems
closed-loop processes, 217, 218
earliest in history, 217
feedback processes, 217–218
open-loop processes, 217, 218
controlling cycle, 35, 36
Cook, S. C., 471
CORBA (common object request broker architecture) standards, 320–323
core cycle, 36
CORE systems engineering tool
classes, 62, 63
documents, 65–66
overview, 62–66
relations, 62–65
Corner, J. L., 399, 407
correct (requirements attribute), 163, 164, 514
cost requirements, 183, 289, 290, 515
Coulouris, G., 262
Cox, C. B., 17, 166
Cox, M. E., 352
Cox, J., 464
Craik, K. J. W., 72
Crowe, D., 146
cybernetics, 448–452, 515
cycle model, 35–36
cycles, in digraphs, 122–123, 133
D
Daly, E., 31
Dam, S., 25
Daniels, J., 160
data flow diagrams (DFDs), 379–382
data link layer, OSI reference model, 315, 317, 319
data models
defined, 375, 515
definition, 4
detailed functions, 37–39
design as decomposition process, 5
definition, 515
detailed functions, 37–39
defined, 70, 515
descriptive models, 70, 71, 515
design
American Heritage Dictionary
definition, 4
descriptive models, 70, 71, 515
design independent (requirements attribute), 163, 164, 515
design structure matrix (DSM), 284, 285, 286
design validity
definition, 330, 515
design
in qualifications chain, 330, 331
role in validation, 331, 333
development decisions, 14, 15
DFDs (data flow diagrams), 379–382
Dickinson, B. W., 219
Dietrich, B. L., 216
digraphs
acyclic, 123
circuits in, 122
comparing with IDEF0 diagrams, 133–135
connectedness in, 123–124
cycles in, 122–123, 133
defined, 118–119, 121–122, 515
paths in, 122
semicircuits in, 123
semicycles in, 123, 133
semiwalks in, 123
trails in, 122
unary relations, 125–127
walks in, 122
directed forests, 132
directed graphs. See digraphs
directed trees, 131–132
discipline engineers, 11, 15, 19, 203, 349
discrete mathematics
functions, 111–113
overview, 100–102
relations, 108–111
sets, 102–108
distributed architecture, 262, 515
distributed computing environment (DCE), 322
distributed systems, 259, 262–264
document class, CORE, 63
documents. See also stakeholders’
requirements document (StkhldrsRD)
IDEF0 model of the engineering of a system, 493–512
outlines, 489–492
system description document (SDD), 65–66, 492
system requirements document (SysRD), 30, 31, 148, 491–492, 523
system requirements validation document (SRVD), 30, 31, 492
DoD architecture framework (DoDAF), 22, 24–25
domain set class, CORE, 63
Dorny, C. N., 219
Douglass, B. P., 486
Drebbel, Cornelis, 217
Driscoll, P. J., 5
Duato, J., 296, 313
Duffy, M. A., 216
Dyer, J. S., 401
E
early validation, 328, 515
edge labeling, in graphs, 121
. . , Edwards, W., 399, 406
EFFBDs (extended function flow block diagrams), 68, 69, 76, 87–89, 91, 99, 17
Elam, J., 399
engineering, defined, 10, 515
engineering of systems
approaches for implementing, 17–22
at Bell Labs, 6–7
comparison of definitions, 8–10
decomposition process, 5
defined, 10, 515
document outlines, 489–492
definitions, 8–10
equivalence, 290, 516
Engstrom, E. W., 6
equivalent functionalities, 222, 231–235
errors, 232, 340, 516
evaluation, 481–483
exit class, CORE, 63
expected utility, 422
external interface requirements, 157–159, 209, 235, 236, 516
external systems diagrams
defined, 516
as key systems engineering concept, 52, 53
external systems diagrams (Continued)
in mobile protected weapons system
application, 430–431
for automated soda machine, 358–361
for stakeholders’ requirements, 171–174

F
FAA wide-area augmentation system
(WAAS) case study, 300–304
Fabrycky, W. J., 20, 183, 298
Fagan, M., 31
Fagen, M. D., 6
failures, 232, 340, 516
Farry, K., 166, 481
Faulk, S., 145–146
fault tolerance
achieving by using redundancy, 266–271
as functional architectural design
goal, 231–235
faults
categorizing, 341
defined, 232, 340, 516
role in qualification process, 340–342
taxonomy of consequences, 341
FBI fingerprint identification system case
study, 257–260, 261
Federal Information Processing Standard
(FIPS) Publications, 80, 87, 314, 376, 396
feedback and control
closed-loop control systems, 217,
218–219
defined, 217, 516
earliest control systems, 217
in functional architecture design,
217–220
IDEF0 illustration, 219, 220
negative feedback process, 218–219
open-loop control systems, 217, 218
positive feedback process, 218–219
Fermi problems, 467–468, 516
Ferrarini, L., 296
FFBDs (function flow block diagrams), 69,
87–94
figure of merit (FOM), 174, 516
fingerprint identification system case
study, 257–260, 261
finite-state machines (FSMs), 387, 392
Finkelstein, A. C. W., 186

FIPS. See Federal Information Processing
Standard (FIPS) Publications
Fitts, Paul, 7, 282
fitts list, 282
flaw of averages, 459
flexibility, defined, 264
flowdown. See requirements flowdown
Flynn, M. J., 263
Folkeson, A., 346
forests, in graph theory, 132
Foroughi, A., 398
Forrester, J. W., 464, 469
Forsberg, K., 5, 9, 10, 37, 39
Fowler M., 486
Frankel, E. G., 298
Franklin, G. F., 219
Frantz, W. F., 163
French, S., 401
Fricke, E., 264, 265
Friedenthal, S., 26, 75, 94
Friedman, G. J., 464, 465, 469
Friend, J., 252, 399
FSMs (finite-state machines), 387
function class, CORE, 63
function flow block diagrams (FFBDs), 69,
87–94. See also EFFBDs (extended
function flow block diagrams)
functional architecture
assigning system’s functions, 210–228
for automated soda machine, 365–367
common mistakes, 230–231
creating functional decomposition,
229–230
decomposition vs. composition, 209–210
defined, 206, 516
defining by using IDEF0 model, 87, 206,
207, 208, 216, 219, 229
developing, 202–238
distinctions between system modes,
states, and functions, 202–203, 204,
205 documenting, 298–299
evaluating hierarchies for shortfalls and
overlaps, 221–228
feedback and control in design, 217–220
hierarchies for development and
manufacturing phases, 216–217
IDEF0 process model overview,
206–209
levels of detail, 206
matching to physical architecture, 245–246
overview, 26–27, 28, 48, 49, 202–206
partitioning functions into subfunctions, 210–216
tracing requirements to elements, 235–237
functional requirements, 50, 51, 56, 157, 158, 182, 517
functionality
complete, 205, 206
defined, 206, 517
simple, 205
functions (engineering)
defined, 203, 516
as key systems engineering concept, 57, 58, 59
mapping to components, 279–288
one-to-one and onto, 246, 248
functions (mathematical)
bijective, 112
composition, 112–113
defined, 516
injective, 112
fundamental objectives, 174, 175, 178, 517
for automated soda machine, 361
fundamental objectives hierarchy, 174–176, 177, 179, 517. See also objectives hierarchy

G
General Motors, 16, 17, 284, 285
general systems theory
defined, 517
generic physical architecture, 242, 243, 244, 245, 246, 248
Gentner, G., 72
Getting, Ivan, 7
Ghahramani, S., 412
Glegg, G. L., 340
Gobinath, P., 286
Goldratt, E. M., 464
Golnaraghi, F., 451
Gomaa, H., 389
Gondran, M., 298
Goode, H. H., 7
Gotel, O. C., 186
Grady, J. O., 145, 147, 164, 290, 328
graphical models, 71–72, 375–397. See also behavior models; data models; process models
graphs, in mathematics. See also digraphs
adjacency in, 124
bipartite, 121–122
comparing with IDEF0 diagrams, 133–135
connectedness in, 123–124
defined, 117, 118–119, 517
isomorphic, 129
Konigsberg bridge problem, 119–120
ordering relations, 127–129
overview, 117–122
reachability in, 124
as trees, 129–132
unary relations, 125–127
Griffin, J. M., 462
Griffith, P. B., 376
guesstimation. See Fermi problems
Guindon, R., 11, 242
Gupta, R., 286

H
Haefele, J. W., 254
Hall, A., 6, 7
hardware redundancy
active, 268–270
defined, 267, 517
hybrid, 270
passive, 267–268
Harel, D., 378, 389, 391
Harker, P. T., 401
Haskins, B., 32, 473
Hasse diagrams, 128, 129, 136, 137, 138, 139
Hastings, D., 160
Hatley, D. J., 20, 211, 212, 214, 216, 229, 256, 380, 392
Hazelrigg, G. A., 428
Hewitt, C., 311
Hickling, A., 252, 399
higraphs, 378–379
Hitchins, D. K., 448, 468
Ho, J., 399
Hoang, N. T., 37
Hogarth, R. M., 414
Holmberg, K., 346
Honour, E. C., 471–472
Hooks, I., 164, 166, 481
Hoppe, M., 342
Howard, R. A., 398, 399, 400
Hubble space telescope, 43–44, 350, 473–474, 483, 484
Human-designed systems, 47, 150, 523
Hunger, J. W., 168, 169

I
ICAM (integrated computer-aided manufacturing), 23, 80
ICOMs (inputs, controls, outputs, and mechanisms), 82, 84, 85, 86, 87, 133, 134, 517
IDEF family of modeling languages, 80–81
IDEF0 (integrated definition for function modeling)
  background, 23, 80–81
  call arrow concept, 87
  defined, 517–518
  exit rules concept, 87
  functional activation rules concept, 87
  in graph theory terms, 133–135
  introduction, 53
  loops concept, 86
  model of the engineering of a system, 493–512
  semantics or elements, 81–82
  shortcomings as modeling process, 80
  in static behavioral process modeling, 80–87
  syntax, diagram, 82–84
  syntax, model, 84–86
  system engineering use, 87
  tunneling concept, 86–87
IDL (interface definition language), 320, 321, 322
In degrees, in graphs, 121
Incidence, in graphs, 121
INCOSE (International Council on Systems Engineering), 9, 148, 175, 186, 457
Influence diagrams, 416–420
information redundancy, 270, 518
information theory, 449, 469, 518
input requirements, 157–158, 165, 181, 182, 235, 236, 237, 518
input/output requirements as category of stakeholder requirement, 157–158, 159, 181–182
defined, 55–56, 518
defining, 63, 65, 157–158, 181–182
deriving for internal items, 289
tracing, 207, 209, 235–237
input/output traces, 146, 170–171, 172, 518
instantiated physical architecture, 244, 246, 247, 248
integrated automated fingerprint identification system (IAFIS), 257–260, 261, 262
integrated computer-aided manufacturing (ICAM), 23, 80
integration
  alternatives to bottom-up process, 337–340
  for automated soda machine, 370, 371
  as big-bang process, 339
  as bottom-up process, 332, 333, 337, 338
  defined, 327, 518
  as engineering consideration, 4
  incremental, 337, 339
  major functions, 333–337
  as middle-out process, 338–339
  overview, 333–337
  phase, 337
  racecar example, 12–14
  asrecomposition process, 5
  at subsystem level, 333–337
  as top-down process, 337, 338
  inVee model, 11–12, 46–47, 50–51
  interconnections, defined, 316
  interface class, CORE, 63
  interface control information, 316
  interface data units (IDUs), 316
  interface definition language (IDL), 320, 321, 322
  interface design, for automated soda machine, 369
  interfaces
    architectures, 310–313
    common object request broker architecture, 319–323
    defined, 518–519
    design process, 323–324
    as key systems engineering concept, 60
    OSI reference model, 315–319
    overview, 307–309
    standards, 313–314
INDEX

Kuo, B. C., 451
Kwinn, M. J., 401

L
Lake, J., 5
Lano, R. J., 23, 166, 382
LaPlace transforms, 449–452
Larsen, R. F., 101–102
law of total probability, 412
layers, OSI reference model, 315, 317–319
Lee, D., 232, 346
Lehnerd, A. P., 265, 475
Lepoeter, K., 37
Levardy, V., 342
Levi, S., 232, 295, 296, 318
Levis, A. H., 25, 26, 27, 118, 150, 286
life cycles
cost commitment and incursion, 8
defined, 519
organizing in StkhdrsRD, 162–163
representation of phases, 5–6, 8
systems, defining, 32–36
as systems engineering consideration, 3–5
Lindley, D., 414
link class, CORE, 63
Lions, J. L., 356
Lister, T., 485
livelock, 296, 519
loops, 86, 119
Lovell, J., 189–190, 191–192

M
Machol, R. E., 7
MacKinnon, D., 315, 316, 318
Magee, C. L., 40–41
Magnuson, E., 272
Manna, Z., 341
Manos, K. L., 147
manufacturing, defined, 519
Mar, B. W., 163
Marca, D. A., 229
Maroni, M., 296
Marshall, C., 181
Martin, J. N., 17, 185
mathematical symbols, 102–103
matrix analysis, 254
Maxwell, D. T., 401
Maxwell, J. C., 217

International Council on Systems Engineering (INCOSE), 9, 148, 175, 186, 457
International Organization for Standards (ISO), 313, 315, 320
International Telephone and Telegraph Consultative Committee (CCITT), 315
intransivity, as property of unary relations, 126, 127
irreflexivity, as property of unary relations, 125
ISO (International Organization for Standards), 313, 315, 320
isomorphisms, 129
issue class, CORE, 63
item class, CORE, 63
items, defined, 58, 519
Jacopini, G., 91
Jagacinski, R. J., 72–73
Jalote, P., 232
Jelassi, M., 398
Johnston, B. W., 267, 268, 269, 270
Johnson, S. B., 17
Johnson-Laird, P., 72
Jones, D. R., 207
Jones, M., 325
Jong, P. d., 311
judgmental models, 71, 72
Karangelen, N. E., 37
Katoen, J. -P., 346
Kauffman, D. L., Jr., 452
Kee, C., 301
Keeney, R. L., 174, 175, 256, 399, 401, 402, 409
Keller, L., 399
Kidder, J. T., 487
Kirkwood, C. W., 399, 403, 407
Kleindorfer, P. R., 414
Klir, G. J., 19
Kluger, J., 189–190, 192
Königsberg bridge problem, 119–120
Kopp, Carlo, 476
network layer, OSI reference model, 315, 317, 319
Newell, A., 242
Nielsen, J., 180, 181, 351
Nii, J., 242
node labeling, in graphs, 121
normative models, 70, 71, 275, 300, 399, 520
N-squared (N^2) diagrams, 23, 87, 382, 383

O
Object Management Group, Inc. (OMG), 23, 75, 319–320, 322
object request broker (ORB) interface, 320–322
objectives hierarchy
adding value curves, 176, 177, 178
for automated soda machine, 361
defined, 175, 520
as key systems engineering concept, 53, 55
in mobile protected weapons system application, 431, 433
for performance requirements, 174–179
object-oriented (OO) design, 22, 23, 246
objects, defined, 22
observance requirements, 160–161, 184, 294, 520
OMG (Object Management Group, Inc.), 75, 319–320
OO (object-oriented) design, 22, 23, 246
open architecture, 263, 520
Open Software Foundation (OSF), 322
open systems, defined, 316
open systems interconnection (OSI) reference model, 315–319
open-loop control processes, 217, 218
operational concept
defined, 520
as key systems engineering concept, 52, 55
in mobile protected weapons system application, 430
for automated soda machine, 357–358
for stakeholders’ requirements, 166–171
operational requirements and tests, 12
operational validity
defined, 74, 328, 520
in qualifications chain, 330, 331
ORB (object request broker) interface, 320–322
order of G, in graphs, 121
Orfali, R., 322
OSF (Open Software Foundation), 322
OSI (open systems interconnection) reference model, 315–319
Ottaway, D. B., 189
out degrees, in graphs, 121
overlap in the functional architecture, 222–228, 520

P
Pages, A., 298
Palmer, J. F., 212
Papaccio, P. N., 21
parametric diagrams
defined, 69
for performance modeling, 96, 97
semantics, 96, 97, 98
Paraskevopoulos, P. N., 450, 451
Parnell, G. S., 401
partially ordered sets, 127, 128, 129
partition on a set A, 107, 520
pathfinder case study, 325–326
paths, in digraphs, 122
Penker, M., 24, 168, 169
Pennington, N., 73
Perdu, D. M., 286
performance analysis, defined, 297–298, 521
performance modeling, in SysML, 96–98
Perry, W. E., 337
Petersen, C. C., 44
Petri nets (PNs), 23, 76, 118, 134, 205, 297, 376, 384, 393, 394–395
Petroski, H., 330
Phan, P., 464, 469
physical architecture
for automated soda machine, 367–369
centralized vs. decentralized, 262
comparison with military’s Work Breakdown Structure, 243–244
creativity techniques, 249–254
physical architecture (Continued)
defined, 241, 521
design flexibility, 257, 264–265
developing, 241–272
development issues, 256–271
development overview, 245–248
distributed, 259, 262
documenting, 298–299
generic vs. instantiated, 242–245
graphic representations, 255–256
matching to functional architecture, 246
morphological box technique, 249–252
option creation techniques, 253–255
overview, 26, 27, 28, 48, 49, 241–242
physical layer, OSI reference model, 315, 317, 319
physical models
defined, 71, 521
vs. qualitative models, 74
pipeline architecture, 312, 313
Pirbhai, I. A., 20, 211, 212, 214, 216, 229, 256, 380, 392
PNs (Petri nets), 23, 76, 118, 134, 205, 297, 376, 384, 393, 394–395
Pohl, E., 183, 298
positive feedback control process, 218–219
Powell, R. A., 476
power set of set A, 107–108, 521
Prang, J., 8
Prasad, B., 8
presentation entities (PEs), 317, 318
presentation layer, OSI reference model, 315, 317, 318
Price, H. E., 282, 283, 284
probability theory, 399–400, 412–414.
  See also uncertainty
problem situation of mission element need
  statement, and systems engineering
  management plan (SEMP), 30, 31
problem-solving techniques, 253–255
process models. See also IDEF0 (integrated
definition for function modeling)
data flow diagrams, 379–382
defined, 69, 375, 521
N-squared charts, 382, 383
overview, 379–383
protocol control information, 316
protocol data units (PDUs), 316
protocols, 316
prototypes, 179, 521
Pugh, S., 401, 405
Q
qualification
  acceptance testing, 349–352
  for automated soda machine, 370, 371
  compared with testing, 327–328
defined, 327, 521
methods categories, 346–349
overview, 340–342
planning levels, 342–346
qualification methods
  analysis and simulation, 346
defined, 521
demonstration, 347
inspection, 346
testing, 346, 347–348, 349
qualification plans, 13–14, 339–340, 342–346
qualification requirements
categories, 294
as category of stakeholder
  requirement, 160–161, 162, 184–185
defined, 521
deriving for components, 294
in qualification planning process, 343–344
tracing, 294
qualitative models
  behavior models, 375, 384–395
data models, 375, 376–379
defined, 72, 521–522
  graphical techniques, 375–397
  process models, 375, 379–383
  vs. quantitative models, 74, 75
quantitative models
defined, 72, 522
vs. mental models, 72–73
vs. qualitative models, 72–73
queueing theory, 457–459, 460–461, 522
R
racecar example, 12
Raiffa, H., 401, 407
Ramo, Simon, 7
RAND Corporation, 7
rank-order centroid (ROC), 406–407, 408
Rasmussen, J., 72
Reason, J., 181
recomposition process, 5, 11–12
redundancy, 266–271
Reed, M. A., 27, 243
reference model for OSI, 315–316
reflexivity, as property of unary relations, 125
regression testing, 337, 482, 522
Reitman, W. R., 242
relations
  binary, 109
  Cartesian products, 108
defined, 522
equivalence, 111
ordered pairs, 108, 109
ordering, 127–129
partial ordering, 111
unary, 109, 110–111, 125–127
relevance diagrams, 414–416
requirement class, CORE, 63
requirements. See also stakeholders’ requirements
  common documents, 29–32
  compared with specifications, 29
defined, 29, 147
  as key systems engineering concept, 55–57
  mission, 29
  modeling in SysML diagrams, 96
  overview, 29–32, 147
  performance, 29
  stakeholder, 29
requirements flowdown, 289–290, 291, 301, 330, 333, 522
requirements statements, 164–165, 522
requirements validity
  defined, 329–330, 522
  in qualifications chain, 330, 331
  role in validation, 331
resource class, CORE, 63
Richardson, D. J., 349
ring architecture, 312, 313
risk, defined, 421, 521
risk analysis, 297, 421, 522
risk avoidance, 421, 522
risk class, CORE, 63
risk issues, uncertainty in, 473–474
risk management, 421, 522
risk mitigation, 483–485
risk preference
  assessing, 422–425
  exponential, 426–427
  overview, 421, 422
  risk reduction, 473
  risk tolerance, 426
  risk transference, 421, 522
Rittel, H., 242
Roberts, R. A., 412
robustness, defined, 264
Rosen, K. H., 101
Ross, A. M., 160
Ross, D., 81
Royce, W. W., 20
Ruh, W. A., 320, 323
Ruijgrok, G. J. J., 462
S
Saaty, T. L., 401, 407, 409
Sabbagh, K.,
SADT (structured analysis and design technique), 23, 81, 229
Sage, A. P., 9, 20, 298
Sagoo, J. S., 453
Sailor, J. D., 9, 147, 177, 179
Samson, D., 346, 348
Sangiorgi, D., 311
Sauser, B., 452, 453, 469
Savage, L. J., 414
Savage, S. L., 459
scenarios, 52–53, 69, 152, 167–171, 522
schedule requirements, 160, 161, 165, 183, 344, 522
Scheiber, S. F., 8
Schkade, D. A., 207
Schlager, K. J., 6
Schmekel, H., 295
Schulz, A. P., 264, 265
Schwartz, M., 315, 318
science and analysis of systems
  aircraft, Breguet range equation, 462–463
  aircraft, quantitative characterization, 462–463, 464
  C-5 aircraft, weight empty
  requirement, 462
causal loop diagrams, 452–453
constraint theory, 464–467
science and analysis of systems (Continued)
cybernetics, 448–452
elevator, quantitative characterization, 454–459
Fermi problems and guesstimation, 467–468
flaw of averages, 459
general systems theory, 446–447
information theory, 449, 469
Laplace transforms, 449–452
natural systems, 448
overview, 445–446
quantitative characterization of systems, 454–464
queuing theory, applied, 457–459, 460–461
soda machine, quantitative characterization, 459–461, 464
system dynamics, 464
systemigrams, 453–454
systems science, 448
systems thinking, 452–454
SDD. See system description document (SDD)
semantics
block definition diagrams, 96–97, 98
block diagrams, 94–96
defined, 69, 76–77, 523
EFFBDS, 87–88
IDEF., 81–82
in models, 70
parametric diagrams, 97, 98
sequence diagrams, 78–79
use case diagrams, 77–78
semicircuits, in digraphs, 123
sycycles, in digraphs, 123, 133
semiwalks, in digraphs, 123
SEMP (systems engineering management plan), 30, 31, 490
Sen, A. K., 428
Senge, P. M., 452, 469
sequence diagrams
elevator example, 78, 79
as input/output traces, 170–171
semantics, 78–79
syntax, 79
service access points (SAPs), 316
service providers, 316
session layer, OSI reference model, 315, 317, 318
sets
defined, 102, 523
descrribing members, 103–104
equality, 105
examples, 103
finite, 104
inclusion, 104
infinite, 104
mathematical symbols, 102–103
operations on, 105–106
partitions, 107
power, 107–108
singleton, 104
special, 104–105
subsets, 104
writing memberships, 103
Shachter, R. D., 414, 417, 419
Shafer, L., 179
Shannon, C., 449, 469
shared memory architectures, 311–312
Shea, Joe, 15, 16–17
Shneiderman, B., 181
shortfall in the functional architecture categories, 221
defined, 221, 523
identifying, 222–228
“showstoppers”, averting, 479–481
Shuey, R. L., 262
Sidewinder missile, 476–477
signal flow graphs, 219
Simon, H. A., 242
simple graphs, defined, 119
simulation models, 71–72
Sinnott, R. W., 44
size of G, 121
Snider, W. D., 407
SofTech, Inc., 81
software engineering
spiral model, 21–22
waterfall model, 20, 28–29
software redundancy, 270, 523
spanning trees, 130–131
specifications
compared with requirements, 29
defined, 29, 523
Spetzler, C. S., 414
spiral model, software engineering, 21–22
Spitzer, Lynn, 43
spoke architecture, 312, 313
SRD (system requirements document), 30, 31, 148, 491–492, 523
SRVD (system requirements validation document), 30, 31, 492
Stael von Holstein, C. A., 414
stakeholders, 15, 523. See also stakeholders’ requirements; stakeholders’ requirements document (StkhldrsRD)
stakeholders’ requirements
air bag restraint system case study, 188–189
analyses, 179–180
Apollo 13 case study, 189–192
for automated soda machine, 362–365
categorizing, 156–157
characteristics, 163–164
defined, 29, 49, 145, 523
defining design problem, 151–155
defining requirements, 181–185
developing, 151–162
external systems diagram, 171–174
input/output requirements, 157–158, 159, 181–182
managing requirements, 185–186
objectives hierarchy for performance requirements, 161, 162, 174–179
operational concept, 166–171
overview, 147–150
prototyping, 179
qualification requirements, 160–161, 184–185
StkhldrsRD, 161–162
structuring using partition, 157–162
system-wide and technology requirements, 158, 182–183
trade-off requirements, 159–160, 183–184
usability testing, 180–181
writing, 164–165
stakeholders’ requirements document (StkhldrsRD)
defined, 523
format overview, 162–163
outline, 490–491
overview, 30, 31
standards, for interface design, 313–314
starvation, 297, 523
Stassen, H. G., 72
state of a system, 203, 204, 523
statecharts, 389–392, 393, 394
stateemode class, CORE, 63
state-transition diagrams (STDs), 387–389
STDs (state-transition diagrams), 387–389
Stevens, A. L., 72
Stevens, R., 185
Stillwell, W. G., 406
strategic check cycle, 35, 36
structured analysis and design technique (SADT), 23, 81, 229
structured programming, comparison to FFBD constructs, 91
subnetworks, defined, 317
Suh, N. P., 50
suitability requirements, 183, 291, 523
surge or race, 297, 523
symbolic models, 71, 72
symbols, mathematical, 102–103
symmetry, as property of unary relations, 126
syntax
block diagrams, 94–96
defined, 69, 77, 523
EFFBDs (extended function flow block diagrams), 87–88
IDEF., 82–86
in models, 70
sequence diagrams, 79
use case diagrams, 77
synthesis, 290, 330, 523
SysML (systems modeling language) compared with TTDSE (traditional, top-down systems engineering), 61–62
defined, 22–23, 69
diagrams for requirements modeling, 96
meta-system and requirements modeling, 76–80
modeling overview, 75–76
overview, 25–26, 61–62
SysRD (system requirements document), 30, 31, 148, 491–492, 524
system context, 47, 150–151, 514, 524
system description document (SDD), 65–66, 492
system dynamics, 464, 524
system, human-designed, 446, 524
system mode, 202–203, 204, 519
system requirements, 49, 100–101, 148, 152, 154, 524
system requirements document (SysRD), 30, 31, 148, 491–492, 524
system requirements validation document (SRVD), 30, 31, 492
system tasks or functions, 47, 150, 524
systemigrams, 453–454
systems. See also engineering of systems
behavior modeling view, 39
data modeling view, 37, 39
defined, 3, 47, 150, 523
design process functions, 47–50
design process overview, 46–51
distinctions between modes, states, and functions, 202–203, 204, 205
environmental modeling view, 37
implementation modeling view, 39
life-cycle considerations, 3–4
modeling views, 37
process modeling view, 39
role of design concept, 4
role of integration concept, 4
stakeholders, 3–4
ways to categorize, 40–41
systems analysis, role of RAND Corporation, 7
systems engineering. See engineering of systems
systems engineering management plan (SEMP), 30, 31, 490
System’s external systems, 47, 150, 524
systems modeling language. See SysML (Systems Modeling Language)
systems science, 448, 524
systems thinking, 452–454, 524
system-wide requirements. See technology and system-wide requirements
as category of stakeholder requirement, 158, 182–183
defined, 524
deriving subsystem-wide requirements, 289–290
tracing, 289–290
Terninko, J., 255
testing. See also acceptance testing
compared with qualification, 327–328
as qualification method, 346, 347–348, 349
textual models, 71, 72
therac-25 case study, 354
Thurston, D. L., 160, 179
time redundancy, 271, 524
top-down integration process, 337, 338
traceable (requirements attribute), 164, 185–186, 187, 524
traced (requirements attribute), 164, 525
trade studies, 298, 525
trade-off requirements
as category of stakeholder requirement, 159–160, 183–184
defined, 525
deriving for components, 291–294
as indirect weight elicitation technique, 407, 410
in mobile protected weapons system application, 431
tracing, 291–294
trails, in digraphs, 122
transivity, as property of unary relations, 125–126
transport control protocol/internet protocol (TCP/IP), 319
transport layer, OSI reference model, 315, 317, 318
trees
defined, 525
directed, 131–132
in graph theory, 129–131
spanning, 130–131
TTDSE (traditional top-down systems engineering)
compared with SysML, 61–62
defined, 18
key terms and basic process, 46–61
layered iterative process, 18–20
modeling approaches, 23
overview, 18–20
as “peeling the onion” process, 353, 354

U
Ulvia, J. W., 407
UML (unified modeling language), 22–23, 23–24. See also SysML (systems modeling language)
unambiguous (requirements attribute), 164, 330, 525
unary relations, 109, 110–111, 125–127, 522
uncertainty in decisions
decision trees, 416–417, 418
expected utility, 421
influence diagrams, 416–420
probability theory, 412–414
relevance diagrams, 414–416
risk preference, 421–427
understandable (requirements attribute), 164, 330, 525
unified modeling language (UML), 22–23, 23–24. See also SysML (systems modeling language)
unique (requirements attribute), 161, 164, 525
U. S. Federal Aviation Administration,
   WAAS case study, 300–304
usability
categorizing users, 181
defined, 525
performance elements, 180
testing, 180
usability testing
defined, 180, 525
as part of verification testing, 351–352
use case diagrams, 69, 77–78
utility curves, 431, 434–436

V
validation. See also conceptual validity;
design validity; requirements validity
compared with verification, 12, 13
comparison with acceptance and verification, 328–333
defined, 12, 327, 525
as key systems engineering concept, 60
relationship to verification, 331
as term in testing and qualification, 327–328
validation plans, 161, 184, 294, 525
value of systems engineering
Black and Decker, 475
communications interface, 478–479
continuous improvement, 485–487
errors, find and fix, 481–483
goal-seeking, 474–477
overview, 471–474
Pioneer 10, 475–476
risk mitigation, 483–485
Sidewinder missile, 476–477
showstoppers, averting, 479–481
successes and failures, 471–472
value propositions, 474–487
value functions, eliciting, 402–403
value weights. See weights
value-focused thinking, 255
Van de Vege, J., 219
Van den Hamer, P., 37
VanGundy, A. B., 253, 254, 255
Vargas, L. G., 401
Vee model, 10–12, 37, 46–47, 50–51
Veldhuyzen, W., 72
verifiable (requirements attribute), 159, 185, 187, 525
verification
compared with validation, 12, 13
comparison with acceptance and validation, 328–333
defined, 12, 327, 525
as key systems engineering concept, 60
overview, 330–331
in qualifications chain, 330, 331
relationship to validation, 331
as term in testing and qualification, 327–328
verification cycle, 35–37
verification plans, 161, 184, 294, 525
verification requirement class, CORE, 63
Verplanck, W. L., 282, 283
Voges, U., 349
Von Braun, Werner, 167
Von Neumann, J., 399
Von Winterfeldt, D., 399

W
Wagenhals, L. W., 25
INDEX

Waldinger, R., 341
Walker, D., 311
Walters, J. M., 179
Walton, M., 160
Warfield, J. N., 19, 468
waterfall model, software engineering, 20, 28–29
Watson, S. R., 179, 398, 402, 406, 407
Watt, James, 217
weights
  analytical hierarchy process
      approach, 407, 408, 409
  balance beam approach, 410–411
  direct elicitation techniques, 405–407
  illustration, 408
  indirect elicitation techniques, 407, 408–411
  in Mobile Protected Weapons System
      application, 434–436
      and utility curves, 435–436
Weinstein, L., 467
Wenzel, S., 35, 36
West, P. D., 242, 399
Westrum, Ron, 477
white box testing, 348, 349, 525
wide-area augmentation system (WAAS)
      case study, 300–304
Wiener, N., 448, 469
Wieringa, R. J., 151
Wiklund, M. E., 180
Wilner, D., 326
Wingard, L., 295
Wolverton, M., 475
Wooldridge, Dean, 7
work breakdown structure (WBS), 243–244
Wright, G., 414
Wymore, A. W., 9, 20, 157, 161, 187, 204, 404, 446
Y
Yager, R. R., 401
Yannakakis, M., 346
Yoon, K., 401
Yourdon, E., 295, 376, 377, 380, 381, 382
Z
Zahavi, R., 320
Zwicky, F., 249