INDEX

A
Abstract class, 22, 164, 258
Abstraction, 258
  in object-oriented systems, 281
Accelerometers, 397, 398
Acceptance tests, 472, 477
Access control, 361
Access control requirements, defined, 445
Acknowledgment messages, 386
Action–object order, 384
Actions
  in activity diagrams, 131, 132, 317
  in behavioral state machines, 222
Action statements, 316
Activities
  in activity diagrams, 131, 132, 317
  in behavioral state machines, 222
black-hole, 136
miracle, 136
Activity-based costing, 94
Activity coordination, 458
Activity diagrams, 119, 120
  actions in, 131, 317
  activities in, 131, 317
control flows in, 131, 317
control nodes in, 132–136
decision node in, 134, 317
final-activity node in, 132–133, 317
final-flow node in, 133, 317
fork node in, 135, 317
guard condition, 134
guidelines for creating, 136–137
initial node in, 132, 317
joint node in, 136, 317
merge node in, 134, 317
object flows in, 131, 317
object nodes in, 131, 317
steps in creating, 137–140
swimlanes in, 136, 317
syntax for, 132, 317
of UML, 316
Activity elimination, 95
Actors
  average, 59
  defined, 59, 206, 217
  in interaction diagrams, 205
  primary, 142
  simple, 59
  specialized, 122
  in use-case diagrams, 121–123
Adjusted usecase points (UCP), 62
Adoption, motivating, 493–495
Aesthetics, 370–371
Aggregation association, 181–183
Aggregation relationships
  in factoring process, 257
  in structural modeling, 166
Agile development, 12–15
  criticisms, 13
  extreme programming (XP), 13–14
  principles, 12
  scrum, 14–15
A-kind-of
  in factoring, 257
  in object-oriented systems, 21
  in structural model, 165
Alexander, Christopher, 170
Algorithm specifications, 316–319
Alpha testing, 477
Alternative flows, 144
Alternative matrix, 274–275, 439
Alternative requirements documentation techniques,
  see Requirements documentation techniques, alternative
Amazon.com, 408, 445
Ambler, S. W., 30, 136, 207, 218, 226
Analysis models, 4
  balancing, 242–257
  evolving into design models, 257–262
  verifying and validating, 242–257
Analysis paralysis, 28
Analysis patterns, 294
Analysis phase, 3–4. See also Systems development
life cycle (SDLC)
  analysis strategy, 3
  requirements gathering, 4
  system proposal, 4
Analysis workflow (Unified Process), 28–29
API, see Application program interface
Appelo, J., 76
Application logic, 419
Application program interface (API), 59, 297, 469
Application service providers (ASP), 270
Application software, 420
Application system, 336–337
Approval committee, 3, 43
Architectural components, 419
Architecture-centric OOSAD, 24
Architecture design, 4
Artifact, 432, 433
As-is system, 3, 86
ASP, see Application service providers
Assemblies, 166
Association
  class, 180
  in communication diagram, 217
Association relationships
  in structural models, 166
  in use-case descriptions, 142
  in use-case diagrams, 122, 123–124
Asymmetric encryption algorithm, 446
Attributes, 165, 204, 282
  derived, 176, 300
  multivalued, 333
  in object-oriented systems, 20
  private, 176
  protected, 176
  public, 176
  visibility of, 176
Attribute sets, 333
Audit files, 330
Augmented reality (AR), 404
Authentication, 446
Authentication requirements, 445–446
  defined, 445
Authentication testing, 477
Authorization testing, 477
Availability and reliability
  requirements, 443–444
Average actors, 59
Avison, D., 91
B
Backyard recycling technique, 431
Bar-code readers, 389
Batch processing, 387–389
Batch reports, 393
Behavior, 282
Behavioral modeling, 202–235
  behavioral state machines for, 221–229
  crude analysis for, 229–232
  interaction diagrams for, 204–221
  verifying and validating, 233–234
Behavioral models, 6, 203
  balancing functional model and, 243–251
  balancing structural model and, 251–254
Behavioral state machine, 221–229, 473
  actions in, 222
  activities in, 222
  creating, 226–229
  elements of, 222–226
  events, 222
  guidelines for creating, 226
  states, 221
  transitions in, 222
Behavioral view, 24
Behavior diagrams, 34
Behaviors, 204
  in object-oriented systems, 20
Bellin, D., 167
Benchmark, 440
Benchmarking, informal, 94
Beta testing, 477
Bias minimization, 393–394
Binding technique, 23
Black-box testing, 471, 472
Black hole states, 226
Booch, Grady, 24, 25, 34
Bottom-up interviews, 98, 99
Brainstorming, 167–169
Breadth of information
  (requirements analysis), 109
Break-even point
  defined, 51
  determination, 47, 50–51
  graphing, 47
Brief description (use cases), 142
Broad and shallow menu, 384
Brynjolfsson, Erik, 429
Business analyst, 18
Business-modeling workflow (Unified Process), 28
Business need, 43, 272
Data Access and Manipulation (DAM), 260
  in factoring, 257
  in object orientation, 282
  in object-oriented systems, 19, 20
  in structural models, 164–165
  utility, 260

Class and method design, 280–321
  constraints, 304–314
  contracts, 304–314
  design criteria, 286–293
  method specification, 314–319
  object design activities, 293–304
  object orientation, characteristics of, 282–286
  verifying and validating, 319–321

Class cohesion, 290
  ideal, 291
  mixed-domain, 291
  mixed-instance, 291
  mixed-role, 291

Class diagrams
  aggregation association, 181–183
  class, 176–179
  generalization association, 181
  object diagrams, 184
  relationships, 179–181
  simplifying, 184

Class library, 297

Class-Responsibility-Collaboration (CRC) cards
  collaborations, 173
  elements of, 173–174
  role-playing, with use cases, 174–175

Class–Responsibility–Collaboration (CRC) cards
  responsibilities, 172

Classroom training, 496

Class tests, 473

Client, 259

Client-based architectures, 420–421

Client computers, 419

Client-server architectures, 421–422

Client-server tiers, 422–424

Closed-ended questions, 97, 98

Cloud computing, 426–428

Cloud services, 270

Clustering, 354–355

Coad, Peter, 286, 289

COBIT (Control Objectives for Information and related Technology), 88

Coding practices, 14

Coding standards, 78

Cognitive map, 405
Cohesion
   class, 290
   defined, 289
   generalization/specialization, 291
   ideal class, 290
   method, 289–290
Collaborations, 173, 258–259
Collectivism, 461
Color
   aesthetics, 370
   cultural meanings of, 407
Columnar data stores, 334
Combo box, 391
Command language, 384
Common object lists, 169
Communication diagrams
   creating, 219
   elements of, 216–218
   examples, 219–221
   guidelines for creating, 218–219
Communication path, 433
Compatibility, 46
Completeness check, 392
Complex actors, 59
Complex systems, 16–17
Component, 297
Computer-aided software engineering (CASE)
   benefits of, 77
   defined, 77
   repository, 77
   tools, 77
Computer-based training (CBT), 496, 497
Concept mapping, 110
Concept maps, 110–112
Conceptual model, 163
Concrete class, 21, 164, 258
Configuration and change management workflow
   (Unified Process), 30, 33–34
Confirmation message, 386
Conflict management, 76
Connascence, 292–293
Consistency, 369, 371–372
Consistency check, 392
Constantine, L. L., 30
Constraints, 304
   types of, 306
Construction, 456–477
   defined, 456
   documentation development, 462–467
   programming management, 457–461
system, 5
   test designing, 467–477
Construction phase (Unified Process), 27
Constructor operation, 176
Container classes, 260
Content awareness, 369–370
Context, 408, 460, 483
Contract, 173, 259, 294, 304
Elements of, 306–314
   fixed-price, 271
   time-and-arrangements, 271
   value-added, 271
Control and security (server-based architecture), 425
Control flows, 131, 132, 317
Controllers, 259
Control nodes
   decision node, 134
   final-activity node, 132–133
   final-flow node, 133
   fork node, 135
   initial node, 132
   joint node, 136
   merge node, 134
Conversion, 485–489. See also specific types of conversion
   location, 486–487
   modules, 487–488
   selecting, 488–489
   style of, 486
Conversion location, 486–487
   phased, 487
   pilot, 487
   simultaneous, 487
   style of, 486
Conversion modules
   modular conversion, 487–488
   whole-system conversion, 487
Conversion strategy
   cost, 489
   risk, 488–489
   time, 489
Conversion style
   direct, 486
   parallel, 486
Costs, 109
   certainty of, 494
   in conversion strategy, 489
   development, 47, 48
   of development, 424–425
   of infrastructure, 424
intangible, 48
materials, 94
operational, 47, 48
in requirements analysis, 109
of transition, 494
Costs and benefits analysis
assigning values to, 48
in change management, 492–493
financial calculations for, 51
identifying, 47–48
Coupling
defined, 286
inheritance, 289
interaction, 287–288
CRC cards, see Class-Responsibility-Collaboration (CRC) cards
Critical path method (CPM), 58
Critical task, 58
Critical thinking skills, 92
CRUDE (create, read, update, delete, or execute), 126
CRUDE analysis, 229–232
CRUDE matrix, 243, 245
Cultural and political requirements
customization requirements, 447
legal requirements, 448
synopsis, 449
Cultural differences, 407–410
Cultural issues, 406–410
and information technology, 483–484
in programming management, 460–461
Cultural requirements, 88
Custom development, 268–269
Customization, 269
Customization requirements, 447

D
DAM classes, see Data Access and Manipulation classes
Data Access and Manipulation (DAM) classes, 260, 338
designing, 357–360
Data access logic, 419
Data access speed optimization, 351–356
clustering, 354–355
denormalization, 351–354
indexing, 355–356
Database, 327
Database and file specifications, 4
Database checks, 392
Database management system (DBMS), 327
Data capture at source, 389390
Data-centered methodology, 5
Data entry operator, 389
Data management layer, 261
Data management layer design, 326–363
data access and manipulation classes, designing, 357–360
mapping problem domain objects to object persistence formats, 337–346
nonfunctional requirements and, 360–361
object persistence formats, 327–337
RDBMS-based object storage, 346–357
verifying and validating, 361–362
Data storage, 419
size of, 356–357
DBMS, see Database management system
Decision node, 132, 134, 317
Decision support systems (DSS), 336
Decomposition, 166
Default value, 390
Delay message, 386
DeMarco, T., 71
Dennis, Alan, 105
Denormalization, 351–354
Dependency
partial, 349
transitive, 351
Dependency relationship, 262, 263
Deployment diagrams, 432–434
Deployment engineering workflow, 27
Deployment workflow (Unified Process), 29, 32
Depth of information (requirements analysis), 109
Derived attributes, 176, 300
Design, 240–275
acquisition strategy, selecting, 273–275
and balancing of analysis models, 242–257
classic, avoiding, 241
custom development, 268–269
evolving analysis models into design models, 257–262
optimization, 298–300
outsourcing, 270–272
packaged software, 269–270
packages and package diagrams, 262–268
restructuring, 297–298
selecting, 272–273
strategies, 268–273
Design models, 29
evolving analysis models into, 257–262
packages and package diagrams, 262–268
Design patterns, 294, 295
Design phase. See also Systems development life cycle (SDLC)
architecture design, 4
database and file specifications, 4
design strategy, 4
program design, 4
Design prototype, 11
Design strategy, 4
Design workflows, 27, 29
Destroy operation, 179
Detail report, 395
Detail use case, 141
Development
costs, 47, 48, 424–425
incremental, 24–25
iterative, 24–25
parallel, 8
phased, 9
waterfall, 7
Digital signatures, 446
Direct conversion, 486
Direct manipulation (navigation control), 385–386
Document analysis, 106–107
Documentation, 78–79
development, 462–467
procedures manuals, 463
reference documents, 463
standards, 78
topics, 463, 465
tutorials, 463
Documentation navigation controls, 463
Documentation structure designing, 463–465
Document data stores, 334
Doing responsibilities, 172
Drop-down list box, 391
Drop-down menu, 385
DSS, see Decision support systems
Duration analysis, 93–94
Dynamic binding, 468
in object-oriented systems, 22–23, 283
Dynamic model, 204

E
Ease of development, 425
Ease of learning, 371
Ease of use, 371
Economic feasibility, 46–51. See also Feasibility analysis
break-even point determination, 50–51
cash flow determination, 48–49
costs and benefits, assigning values to, 48
costs and benefits, identifying, 47–48
net present value (NPV), 49–50
return on investment (ROI), 50
Edit checks, 391
EIS, see Executive information systems
E-JAD, see Electronic JAD
Elaboration phase (Unified Process), 27
Electronic brainstorming, 169
Electronic distribution, 104
Electronic JAD, 102
Encapsulation, 282, 468
in object-oriented systems, 20–21
in testing and object orientation, 468
Enchanted objects, 429
Encryption, 445–446
defined, 445
End-user DBMS, 327
Engineering workflows. See also Workflows
analysis workflow, 28–29
business-modeling workflow, 28
deployment workflow, 29
design workflow, 29
implementation workflow, 29
requirements workflow, 28
testing workflow, 29
English-language messages, 406
Enhanced Unified Process, 31, 33
Enterprise DBMS, 327
Enterprise resource planning (ERP), 269
Environmental factors (EF), 59, 62
Environmental factor value (EFactor), 59, 62
Environment and infrastructure management, 76–79.
See also Project management
CASE tools, 77
documentation, 78–79
standards, 77–78
Environment workflow (Unified Process), 30, 32
Error(s), 153
Error correction, 153
Error message, 386
Essential use case, 141, 372
Estimates, refining, 69–70
Estimation, defined, 58
Event
in behavioral state machines, 222, 223
in method specification, 314
Event driven languages, 314
Evolutionary work breakdown structures, 63–67
e-waste, 431
Exceptional flows, 144
Execution occurrence, 206
Executive information systems (EIS), 336
Extend relationship, 122, 144
Extent, 332
External nonfunctional dimensions, 88
External trigger, 142
Extreme programming (XP), 13–14

F
Facilitator, 101, 104
Factoring, 257–258, 298
Familiarity
  with functional area, 46
  with technology, 46
Fan-out, 300
Fat client, 421
Faults, 153
Feasibility analysis, 3, 43, 45–53. See also Project management
  economic feasibility, 46–51
  organizational feasibility, 51–53
  technical feasibility, 45–46
Feminine cultures, 409
Field labels, 370
Final-activity node, 132–133, 317
Final-flow node, 132, 133, 317
Final state, 222, 223
Financial awards, 75
First-line supervisors, 492
First mover, 43
First normal form (1NF), 347, 349
Fitzgerald, G., 91
Fixed-price contract, 271
Flow of events
  alternative or exceptional flows, 144
  normal, 144
  subflows, 144
  in use-case description, 144–145
Foreign key, 330, 331
Fork node, 132, 135, 317
Formal usability testing, 381–382
Format check, 392
Foundation layer, 260
Frame, 209, 217, 223
Framework, 297
Frequently asked questions (FAQ), 498
Friedman, T. L., 89, 407
Functional decomposition, 144
Functionality, 44
Functional lead, 74

G
Games, 400
Gamification, 400–401
Gantt chart, 56–57
Generalization association, 181
Generalization relationship, 122, 144, 165, 257
Generalization/specialization cohesion, 291
Generic sequence diagram, 204
Globalization, 89
Glocalization, 407
Gradual refinement, 3
Grammar order, consistent, 384
Graphical displays and reports, 393
Graphical user interface (GUI), 368, 425
Graphs, 395
Green data centers, 431
Green IT, 431–432
Grid computing, 426
Ground rules (JAD sessions), 103
Group cohesiveness, 76
Guard condition, 134, 217, 222

H
Hall, Edward, 407, 408, 409, 460, 483, 484
Haptic feedback, 396
Hardcoded value, 471
Hardware and operating system, 360
Hardware and software specification, 438–440
Hardware components, primary, 419
Has-parts
  in factoring, 257
  in structural model, 166
Health and Human Services Health Insurance Portability and Accountability Act (HIPAA), 428
Help desk, 498
Help message, 386
Heuristic evaluation, 381
Index

History files, 330
Hofstede, Geert, 407, 408, 409, 460, 461, 483, 484
Holland, Ian M., 287
Hot keys, 385
Human-computer interaction layer, 261
Human-computer interaction layer design, 367–410
  games, 400–402
  gamification, 400–402
  immersive environments, 404–406
  input design, 387–392
  international and cultural issues, 406–410
  mobile computing, 395–398
  multidimensional information visualization design, 402–404
  navigation design, 383–387
  nonfunctional requirements and, 410
  output design, 392–395
  social media and, 398–400
  user interface design, 368–372, 395–398, 400–402
  user interface design process, 372–383
Hybrid clouds, 426

I

Ideal class cohesion, 290
Image map, 385
Immersive environments, 404–406
Impedance mismatch, 336
Implementation phase, 4–5. See also Systems
development life cycle (SDLC)
  construction, 5
  installation, 5
  support plan, 5
Implementation workflow (Unified Process), 29
Importance level (use-cases), 142
Inception phase (Unified Process), 26–27
Incidents, 169
Include relationship, 122, 144
Incremental development, 24–25
Indexing, 355–356
Individualism, 461
  versus collectivism, 409, 484
Informal benchmarking, 94, 147
Informational strategy, 493
Information hiding, 282
  in object-oriented systems, 20–21
  in testing and object orientation, 468
Information load, 393
Infrastructure analyst, 19
Infrastructure as a Service (IaaS), 427
Infrastructure cost, 424
Infrastructure design, 432–438
  deployment diagrams, 432–434
  network model, 434–438
Infrastructure management workflow (Unified Process), 32
Inheritance, 144, 469
  conflict, 284, 285
  multiple, 285
  in object orientation, 284–286
  in object-oriented systems, 21–22
  single, 284
Inheritance coupling, 289
In-house experience, 272
Initial node, 132, 317
Initial state, 222, 223
Input design, 387–392
  basic principles, 387–390
  input validation, 391–392
  types of inputs, 390, 391
Input validation, 391–392
Installation process, 5, 481–501
  change management, 489–497
  conversion, 485–489
  cultural issues in, 483–484
  post-implementation activities, 497–501
Instance sequence diagrams, 204
Instantiation, 184
Institutionalization, 497
Intangible benefits, 47, 48
Intangible costs, 48
Intangible value, 44
Integration of information, 109
Integration testing, 468, 472
Integration tests, 475–476
Interaction, 169
Interaction coupling, 287–288
Interaction diagrams, 204–221
  communication diagrams, 216–221
  messages, 204
  objects in, 204
  operations in, 204
  sequence diagrams, 204–215
Interaction testing, 472, 475
Interactive evaluation, 381
Interface actions, 377
Interface capabilities, 425
Interface design, 4
Interface design prototyping, 377–380
  selecting, 379
  storyboard, 377–379
J

Jacobson, Ivar, 24, 25, 34

Jelled team, 71–72

Join node, 132, 317

Joint application development (JAD), 100–104, 105

conducting session, 103–104
designing, 103
electronic, 102

ground rules, 103

participant selection, 102–103

post-session report, 104

preparing for session, 103

problem management in, 105

for RAD-based methodologies, 9

Joint node, 136

Jones, Capers, 101

Karner, Gustav, 58

Keystrokes minimization, 390

Key-value data stores, 334

KISS principle, 13, 146

Knowing responsibilities, 172

Krug, Steve, 382, 383, 396, 408

K

Karner, Gustav, 58

Keystrokes minimization, 390

Key-value data stores, 334

KISS principle, 13, 146

Knowing responsibilities, 172

Krug, Steve, 382, 383, 396, 408

L

Languages (navigation control), 384

Larman, C., 15, 147

Law of Demeter, 287, 288

Layers, 259–262. See also Design
data management, 260–261

foundation, 260

human–computer interaction, 261

physical architecture, 261–262

problem domain, 260

Layout (user interface design), 369

Legal requirements, 448

Lencioni, P., 72

Lewin, Kurt, 482

Lieberherr, Karl J., 287

Lifeline, 205, 206

Linked list, 328

Lister, T., 71

Load tests, 477

Local area network (LAN), 420

Locations, 435

Logical models, 120

Long-versus short-term orientation, 461, 484

Lookup files, 328

M

Magnetic stripe readers, 389

Maintainability requirements, 441, 442

Maintenance oracle, 154
Index

Management information systems (MIS), 336
Management policies, 491–492
Manual systems, 89
Masculinity versus femininity, 409, 484
Master files, 328
Materials costs, 94
McAfee, Andrew, 429
McEwen, Adrian, 429
Measurements, 492
Media, 394–395
Meeting, scrum, 15
Menu bar, 385
Menus (navigation control), 384–385
Merge node, 132, 134, 317
Message passing, 315
Messages
  defined, 206, 217
  in interaction diagrams, 204
  navigation design, 386–387
  in object orientation, 282
  in object-oriented systems, 20
Method(s), 165, 204, 257, 259
  in object orientation, 282
  in object-oriented systems, 20
Method cohesion, 289–290
  classical, 290
  coincidental, 290
  communicational, 290
  functional, 290
  logical, 290
  procedural, 290
  sequential, 290
  temporal, 290
Methodology(-ies), 5–17
  agile development, 12–15
  criteria for selecting, 15–17
  data-centered, 5
  defined, 5
  object-oriented, 5
  process-centered, 5
  rapid application development (RAD), 8–12
  sequencing of SDLC phases, 5
  structured design, 6–8
Method specification, 314–319
  algorithm specifications, 316–319
  events, 314
  general information, 314
  message passing, 315
Meyers, Glenford, 290
Middle managers, 492
Middleware, 421
Migration plan, 482
Milestones, project, 55, 57
Miracle states, 226
Mission-critical systems, 445
  agile for, 13
  and Scrum, 15
  XP for, 14
Mistakes
  implementation, 459
  preventing, 383
  recovery from, 383
Mobile computing, 395–398
Mobile devices, 396, 439
Model–View–Controller (MVC)
  architecture, 259
Modular conversion, 487–488
Module, 257
Monochronic time, 408, 461, 484
Motivation, 75–76
Multidimensional information visualization
  design, 402–404
Multilingual requirements, 406–407
Multiple inheritance, 285
Multiple layout, 369
Multiplicity, 180
Multitenancy, 426
Multivalued attributes, 333
MVC architecture, see Model-View-Controller
  (MVC) architecture
N
Narrow and deep menu, 384
Natural language, 384
Navigation controls
  consistency in, 371
  direct manipulation, 385–386
  languages, 384
  menus, 384–385
Navigation design, 383–387
  basic principles, 383–384
  documentation, 387
  grammar order, consistent, 384
  messages, 386–387
  preventing mistakes, 383
  recovery from mistakes, 383
  types of controls, 384–386
Navigation terms identification, 465–467
Net present value (NPV), 47, 49–50
  defined, 51
Network, 419
Network diagram, 57–58
Network model, 434–438
Node, 58, 432, 433
Nonfunctional requirements, 87, 88
   cultural, 88, 90
   cultural and political requirements, 447–448
   and data management layer design, 360–361
   and human-computer interaction layer design, 410
   operational, 88, 90
   operational requirements, 441–442
   performance, 88, 90
   performance requirements, 442–444
   and physical architecture layer design, 440–449
   political, 88, 90
   security, 88, 90
   security requirements, 444–447
   synopsis, 448–449
Normal flow of events, 144
Normalization process, 298, 347
NoSQL data stores, 333–334, 335
n-tiered architecture, 422
   advantage of, 423
   disadvantage of, 424
Null values, 347
Number box, 390

O

Object(s)
   defined, 206, 217
   in interaction diagrams, 204, 205
   in object orientation, 282
   in object-oriented systems, 19, 20
   temporary, 205
Object–action order, 384
Object-based language, 301–303
Object Constraint Language (OCL), 304, 305
Object design activities, 293–304
   adding specifications, 293
   mapping problem-domain classes to implementation languages, 300–304
   opportunities for reuse, 294–297
   optimizing design, 298–300
   restructuring design, 297–298
Object diagrams, 184
Object flows, 131, 132, 317
Object identification
   brainstorming for, 167–169
   common object lists for, 169
   patterns for, 169–172
   in structural modeling, 166–172
   textual analysis for, 166–167
Object Management Group (OMG), 34, 119
Object nodes, 131, 132, 317
Object orientation
   classes, 282
   dynamic binding, 283
   encapsulation, 282
   information hiding, 282
   inheritance, 284–286
   messages, 282
   methods, 282
   objects, 282
   polymorphism, 282–284
   and testing, 468–469
Object-oriented database, 332–333, 335
Object-oriented database management systems (OODBMS), 332
   mapping problem domain objects to, 338–341
Object-oriented development process and products, 469
Object-oriented methodology, 5
Object-oriented programming language (OOPL), 333
Object-oriented systems
   attributes in, 20
   behaviors in, 20
   classes in, 19, 20
   dynamic binding in, 22–23
   encapsulation in, 20–21
   information hiding in, 20–21
   inheritance in, 21–22
   messages in, 20
   methods in, 20
   objects in, 19, 20
   polymorphism in, 22
Object-oriented systems analysis and design (OOSAD), 23–25
   architecture-centric, 24
   benefits of, 25
   incremental development, 24–25
   iterative, 24–25
   use-case driven, 24
Object persistence formats
   application system, type of, 336–337
   criteria for files, 337
   data types supported, 336
   future needs, 337
   mapping problem domain objects to, 337–346
   NoSQL data stores, 333–334, 335

Index 517
Object persistence formats (continued)
    object-oriented database, 332–333, 335
    object-relational databases, 332, 335
    random access files, 328, 335
    relational database, 330–332
    selecting, 335–337
    sequential access files, 327, 335
    storage formats, existing, 337
    strengths of files, 335
    weaknesses of files, 335–336
Object recognition, 404
Object-relational database management systems (ORDBMS), 332
    mapping problem domain objects to, 341–344
    using DAM classes, 358
Object-relational databases, 332, 335
Object storage optimization, RDBMS-based, 346–357
    data access speed, optimizing, 351–356
    data storage size, 356–357
    storage efficiency, optimizing, 347–351
Object wrapper, 270
Observation, 108
Occlusion, 403
OCL, see Object Constraint Language
OMG, see Object Management Group
On-demand training, 498
One-on-one training, 496
Online documentation, 463
Online support, 498
Online versus batch processing, 387–389
On-screen list box, 391
OODBMS, see Object-oriented database management systems
OOPL, see Object-oriented programming language
OOSAD, see Object-oriented systems analysis and design
Open-ended questions, 97, 98
OPEN process (Object-oriented Process, Environment, and Notation), 31
Operating system, 438
Operation, 165, 204
    constructor, 176
    destructor, 176
    query, 176
    update, 176
Operational costs, 47, 48
Operational requirements, 88, 360, 410
    maintainability requirements, 441, 442
    portability requirements, 441, 442
    synopsis, 448
    system integration requirements, 441, 442
    technical environment requirements, 441–442
    Operation call messages, 207
    Operations and support workflow (Unified Process), 32
    Optical character recognition, 389
    ORDBMS, see Object-relational database management systems
    Ordered sequential access files, 328
    Organizational feasibility, 51–53
    Organizational management, 52
    Outcome analysis, 95
    Output design, 392–395
        basic principles, 392–394
        media, 394–395
        types of output, 394
    Outsourcing, 270–272
    Overview information, 142
    Overview use case, 141
    Package(s)
        in class diagrams, 184
        communication diagram, 218
        in design model, 262–268
        in use-case diagram, 127
    Package diagrams
        creating, 266
        dependency relationship in, 262, 263
        in design model, 262–268
        guidelines for creating, 264–265
        syntax for, 263
        verification and validation of, 266–268
    Packaged software, 269–270
    Page-Jones, Meilir, 291
    Paper-based documentation, 462
    Paperless office, 432
    Parallel conversion, 486
    Parallel development, 8
    Parallelization, process, 94
    Parkinson’s Law, 440
    Partial dependency, 349
    Partitions, 258–259
    Patterns, 294
        for object identification, 169–172
    Perceived benefits, 491
    Perceived costs, 491
    Pereira, Arun, 407, 408
    Performance requirements, 88, 360, 410
        availability and reliability requirements, 443–444
        capacity requirements, 443
        speed requirements, 442, 443
        synopsis, 448
Performance testing, 477
Person, 164, 165
Person-hours multiplier (PHM), 63
Phase(s)
  construction, 27
  elaboration, 27
  inception, 26–27
  production, 31–32
  transition, 27–28
  of Unified Process, 26–28
Phased conversion, 487, 489
Phased development, 9, 10
Physical architecture layer, 261–262
Physical architecture layer design, 418–449
  architectural components, 419
  client-based architectures, 420–421
  client–server architectures, 421–422
  client–server tiers, 422–424
  cloud computing, 426–428
  Green IT, 431–432
  hardware and software specification, 438–440
  infrastructure design, 432–438
  Internet of Things (IoT), 428–431
  nonfunctional requirements and, 440–449
  selecting, 424–425
  server-based architectures, 420
  ubiquitous computing, 428
  verifying and validating, 449
Physical models, 120
Pilot conversion, 487, 489
Pink, D. H., 74, 76
Planning phase. See also Systems development life cycle (SDLC)
  project initiation, 3
  project management, 3
Platform as a Service (PaaS), 427
Pointer, 328
Political and cultural requirements, 361
Political requirements, 88
Political strategy, 494, 495
Polychronic time, 408, 461, 483
Polymorphism, 22, 282–284, 468
Pop-up menu, 385
Portability requirements, 441, 442
Portfolio management, 53
Postcondition, 306
Post-implementation activities, 482
  project assessment, 500–501
  system maintenance, 498–500
  system support, 497–498
Post-session report (JAD session), 104
Potential adopters, 490
Power distance, 408, 484
Precondition, 306
Presentation logic, 419
Presenters, 153
Present value (PV), defined, 51
Primary actor, 142
Primary insurance carrier, 179
Primary key, 330, 349
Private attribute, 176
Private clouds, 426
Probing question, 97, 98
Problem analysis, 92
Problem-domain classes to implementation languages, mapping, 300–304
  in object-based language, 301–303
  in single-inheritance language, 301
  in traditional language, 304
Problem domain layer, 260
Problem domain models, 120
Problem domain objects to object persistence formats, 337–346
  to OODBMS format, mapping, 338–341
  to ORDBMS format, mapping, 341–344
  to RDBMS format, mapping, 344–346
Problem management (JAD sessions), 105
Problem report, 498
Procedural standards, 78
Procedures manuals, 463
Process-centered methodology, 5
Process integration, 94
Process models, 120
Process parallelization, 94
Production phase (Unified Process), 31–32
Program design, 4
Program Evaluation and Review Technique (PERT), 57–58
Program log, 458
Programmers, 18, 457–461
Programming management
  activity coordination, 458
  cultural issues, 460–461
  programmers, assigning, 457–458
  schedule management, 458–459
Project, 42
Project assessment, 497
  project team review, 500–501
  system review, 501
Project binder, 78
Project charter, 76
Project effort estimation, 58–63
Project identification, 43–45
  system request, 44
Project initiation, 3
Project management, 3, 41–80, 42, 273
  environment and infrastructure management, 76–79
  feasibility analysis in, 45–53
  project effort estimation, 58–63
  project identification in, 43–45
  project selection, 53–54
  staffing in, 71–76
  traditional tools for, 54–58
  workplan, creating/managing, 63–71
Project management tools, traditional, 54–58
Gantt chart, 56–57
  network diagram, 57–58
  work breakdown structure (WBS), 55–56
Project management workflow (Unified Process), 29–30, 33
Project manager, 3, 19, 42
Project plan, 3
Project size, 46
Project skills, 272–273
Project sponsor, 3, 42, 43
Project team(s), 274
Project team review, 500–501
Protected attribute, 176
Prototyping, 9–11
  throwaway, 11–12
Public attribute, 176
Public clouds, 426
Public key, 446
Public key infrastructure (PKI), 446
Pull approaches (social media), 399
Push approaches (social media), 399

Query operation, 178
Questionnaires, 104–106
  administration of, 106
  designing, 105
  participants selection, 104–105

RAD, see Rapid application development
Radio button, 391
Random access files, 328, 335
Range check, 392
Rapid application development (RAD), 8–12
  phased development, 9
  prototyping, 9–11
  throwaway prototyping, 11–12
Rational Software, 34
Raw data, 356
RDBMS, see Relational database management systems
Ready adopters, 495
Real benefits, 491
Real costs, 491
Real-time reports, 393
Real use case, 141, 373, 387
Recorders, 153
Redefinition, 284, 285
Reference documents, 463
Referential integrity, 330, 331
Refinement
  in factoring, 258
Regular meetings, 458
Reich, Robert, 74
Relational database, 330–332
Relational database management systems (RDBMS), 330
  data access speed, optimizing, 351–356
  data storage size, 356–357
  mapping problem domain objects to, 344–346
  referential integrity, referencing, 330
  storage efficiency, optimizing, 347–351
Relationships
  aggregation, 166
  association, 142, 166
  extend, 144
  generalization, 144, 165–166
  include, 144
  sets, 333
Reliability, system, 17
Reluctant adopters, 495
Repeating groups (fields), 333
Reporting structure, 73
Report usage, 393
Request for information (RFI), 274
Request for proposal (RFP), 274
Request for quote (RFQ), 274
Requirements. See also specific types of requirements
  business, 44, 87
  functional, 87
  gathering, 4
  system, 87
Requirements analysis strategies, 92–95
  activity-based costing, 94
  activity elimination, 95
  duration analysis, 93–94
informal benchmarking, 94
outcome analysis, 95
problem analysis, 92
root cause analysis, 92–93
technology analysis, 95
Requirements determination, 86–91
defining requirement, 87–89
determining requirements, 89–91
purpose, 87
real-world problems with, 91
requirements definition creation, 91
requirements definition report, 89
Requirements documentation techniques, alternative
concept maps, 110–112
user stories, 112
Requirements-gathering techniques, 95–110
combining, 109–110
document analysis, 106–107
interviews, 96–100, 101
joint application development (JAD), 100–104
observation, 108
questionnaires, 104–106
selection of, 108–110
Requirements workflow (Unified Process), 28
Resistance to change, 490–491
Resistant adopters, 495
Resource allocation, 492
Responsibilities
CRC cards, 172
doing, 172
knowing, 172
Return message, 207
Return on investment (ROI), 47, 50
defined, 51
Reuse, 469
Rewards, 492
RFI, see Request for information
RFP, see Request for proposal
RFQ, see Request for quote
Risk
assessment, 70, 71
in conversion strategy, 488–489
management, 70–71
Role-playing CRC cards, 110, 141
with use cases, 174–175
Root cause analysis, 92–93
Rose, David, 430
Round-robin approach, 169
Round-trip engineering, 77
Rumbaugh, James, 24, 25, 34
Structural model, 163, 164
    balancing behavioral model and, 251–254
    balancing functional models and, 242–243
    and functional model, relationships, 244
Structural modeling, 163–197
    attributes, 165
    class diagrams, 176–185
    classes, 164–165
    CRC cards, 172–175
    creating, 185–194
    object identification, 166–172
    operations, 164–165
    primary purposes of, 164
    relationships, 165–166
    verifying and validating, 194–197
Structured design, 6–8
    parallel development, 8
    waterfall development, 7
Structured English, 316
Structured interviews, 98
Structured query language (SQL), 332, 419
Stubs, 471
Subclass
    in behavioral state machines, 225
    in generalization relationships, 165
    in object-oriented systems, 21
Subflows, 144
Subject boundary, 122, 125–126
Subject–Verb–Direct-Object–Preposition–Indirect object (SVDPI), 186
Submenus, 384
Substitutability, 166
Summary report, 395
Support plan, 5
Swimlane, 132, 136, 317
Symmetric encryption algorithm, 446
Synopsis
    cultural and political requirements, 446
    operational requirements, 448
    performance requirements, 448
    security requirements, 449

Smart cards, 389
Snyder, Alan, 289
Social media, 398–400
Social networking platforms, 261
Software as a Service (SaaS), 427
Software quality, 88
Software testing, 467
SOP, see Standard operating procedures
Source data automation, 389
Space, 370
Special issues, 44
Specialized actor, 122
Specification requirement standards, 78
Speed of messages, 408, 461, 483
Speed requirements, 442, 443
Sponsors, 489, 490
SQL, see Structured query language
Staffing, 71–76. See also Project management
    conflict management, 76
    jelled team, 71–72
    motivation, 75–76
    staffing plan, 73–74
Stakeholder analysis, 52
Stakeholders, 142
Standard operating procedures (SOP), 492
Standards
    coding, 78
    documentation, 78
    environment and infrastructure management, 77–78
    procedural, 78
    specification requirement, 78
    user interface design, 78
State, 221
    black hole, 226
    defined, 222, 223
    final, 222
    initial, 222
    miracle, 226
State symbol, 222
Static binding, 23
Static model, 176
Static structure diagram, 184
Steering committee, 3
Stereotype, 375, 432
Storage efficiency, optimizing, 347–351
Storage formats, 337
Storyboard, 377–379
Story cards, 112
Strategic alignment, 52
Stress tests, 477
System complexity, 16–17
System documentation, 462
System integration requirements, 441, 442
System interface testing, 475
System maintenance, 497, 498–500
System proposal, 4, 113
System reliability, 17
System request, 3, 42, 44, 45, 499
System requirements, 87
System review, 501
Systems analyst, 18
  business analyst, 18
  change management analyst, 19
  infrastructure analyst, 18–19
  primary objective of, 2
  project manager, 19
  roles and skills, 17–19
Systems development life cycle (SDLC), 2–5
  analysis phase, 3–4
  defined, 1
  design phase, 4
  implementation phase, 4–5
  planning phase, 3
Systems integration, 270
System specification, 4
System support, 497–498
System tests, 472, 476–477
System users, 52, 53
System value, 444–445
System value estimates, defined, 445

T

Table scan, 354
Tab menu, 385
Tangible benefits, 47, 48
Tangible value, 44
Task, 54
Task information, 54
Task lists, 112
Teams
  autonomy for, 76
  complexity with, 73
  dysfunctional, 72
  jelled, 72
  leaders of, 76
  scrum, 15
Technical complexity factors (TCF), 59, 62
Technical environment requirements, 441–442
Technical factor value (TFactor), 59, 62
Technical feasibility, 45–46
Technical lead, 74
Technical risk analysis, 46
Technical skills, 74
Technical writer, 18
Technology
  analysis, 95
  familiarity with, 16
Temporal trigger, 142
Temporary object, 205
Testing workflow (Unified Process), 29
Test planning, 469–471
Tests, designing, 467–477
  acceptance tests, 477
  integration tests, 475–476
  and object orientation, 468–469
  system tests, 476–477
  test planning, 469–471
  unit tests, 471–475
Test specifications, 471
Test workflow (Unified Process), 32
Text box, 390
Textual analysis, 166–167
Thick client, 421
Thin client, 421
Third normal form (3NF), 351
Three-tiered architecture, 422
Throwaway prototyping, 11–12, 110
Tidwell, Jenifer, 396, 397
Time-and-arrangements contract, 271
Timeboxing, 68–69
Time dimension, 461
Time frame, 273
Time in conversion strategy, 489
Timesharing, 427
To-be system, 3, 86
Tool bar, 385
Top-down interviews, 98, 99
Total cost of ownership, 422
Touchscreen, 397
Traceability of artifacts, 457
Trade-offs, 54
Traditional language, 304
Training (Change management), 495–497
Training plan, 5
Transaction files, 328
Transaction processing, 336, 387
Transition phase (Unified Process), 27–28
Transition process, 222, 223, 375, 491
Transitive dependency, 351
**Index**

<table>
<thead>
<tr>
<th>Trigger</th>
<th>overview information in, 142</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>process documentation with, 140–152</td>
</tr>
<tr>
<td></td>
<td>relationships, 142–144</td>
</tr>
<tr>
<td>Turnaround document</td>
<td>395</td>
</tr>
<tr>
<td>Tutorials</td>
<td>463</td>
</tr>
<tr>
<td>Two-tiered architecture</td>
<td>422</td>
</tr>
<tr>
<td>Type of information</td>
<td>(requirements analysis), 108–109</td>
</tr>
</tbody>
</table>

**U**

<table>
<thead>
<tr>
<th>Ubiquitous computing</th>
<th>428–431</th>
</tr>
</thead>
<tbody>
<tr>
<td>UML, see Unified Modeling Language</td>
<td>34–36, 119</td>
</tr>
<tr>
<td>Unadjusted Actor Weight Total (UAW), 59</td>
<td></td>
</tr>
<tr>
<td>Unadjusted use-case points (UUCP), 59</td>
<td></td>
</tr>
<tr>
<td>Unadjusted use-case weight total (UUCW), 59</td>
<td></td>
</tr>
<tr>
<td>Uncertainty avoidance, 409, 484</td>
<td></td>
</tr>
<tr>
<td>Unified Modeling Language (UML), 34–36, 119</td>
<td></td>
</tr>
<tr>
<td>objective of, 34</td>
<td></td>
</tr>
<tr>
<td>UML 2.5 diagram summary, 35</td>
<td></td>
</tr>
<tr>
<td>Unified Process, 25–34</td>
<td></td>
</tr>
<tr>
<td>documentation in, 79</td>
<td></td>
</tr>
<tr>
<td>enhanced, 31, 33, 64</td>
<td></td>
</tr>
<tr>
<td>extensions to, 30–34</td>
<td></td>
</tr>
<tr>
<td>phases, 26–28</td>
<td></td>
</tr>
<tr>
<td>workflows, 28–30</td>
<td></td>
</tr>
<tr>
<td>Unit tests, 471–475</td>
<td></td>
</tr>
<tr>
<td>Unordered sequential access file, 328</td>
<td></td>
</tr>
<tr>
<td>Unstructured interviews, 98</td>
<td></td>
</tr>
<tr>
<td>Update anomaly, 347</td>
<td></td>
</tr>
<tr>
<td>Update operation, 179</td>
<td></td>
</tr>
<tr>
<td>Usability testing, 476</td>
<td></td>
</tr>
<tr>
<td>Use case, 24, 120</td>
<td></td>
</tr>
<tr>
<td>behavioral models, 203</td>
<td></td>
</tr>
<tr>
<td>complex, 59</td>
<td></td>
</tr>
<tr>
<td>defined, 59, 120</td>
<td></td>
</tr>
<tr>
<td>detail, 141</td>
<td></td>
</tr>
<tr>
<td>essential, 141</td>
<td></td>
</tr>
<tr>
<td>identifying, 126–127</td>
<td></td>
</tr>
<tr>
<td>overview, 141</td>
<td></td>
</tr>
<tr>
<td>real, 141</td>
<td></td>
</tr>
<tr>
<td>role-playing CRC cards with, 174–175</td>
<td></td>
</tr>
<tr>
<td>simple, 59</td>
<td></td>
</tr>
<tr>
<td>testing, 472, 475</td>
<td></td>
</tr>
<tr>
<td>types of, 141</td>
<td></td>
</tr>
<tr>
<td>in use-case diagrams, 122, 124</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use-case description</th>
<th>121</th>
</tr>
</thead>
<tbody>
<tr>
<td>creating, 146–152</td>
<td></td>
</tr>
<tr>
<td>elements of, 141–145</td>
<td></td>
</tr>
<tr>
<td>flow of events, 144</td>
<td></td>
</tr>
<tr>
<td>guidelines for creating, 145–146</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User effort minimization</th>
<th>372</th>
</tr>
</thead>
<tbody>
<tr>
<td>User involvement, 109</td>
<td></td>
</tr>
<tr>
<td>User participation, 53</td>
<td></td>
</tr>
<tr>
<td>User requirements, clarity of, 16</td>
<td></td>
</tr>
<tr>
<td>User stories, 112</td>
<td></td>
</tr>
<tr>
<td>Use scenarios, 372</td>
<td></td>
</tr>
<tr>
<td>Utility classes, 260</td>
<td></td>
</tr>
</tbody>
</table>
Validation
  of analysis models, 242–257
  of behavioral models, 233–234
  of class and method design, 319–321
  of data management layer design, 361–362
  of functional modeling, 154–156
  of package diagrams, 266–268
  of physical architecture layer design, 449
  of structural modeling, 194–197
Validation of input, 391–392
Value-added contract, 271
Verification
  of analysis models, 242–257
  of behavioral models, 233–234
  of class and method design, 319–321
  of data management layer design, 361–362
  of functional modeling, 154–156
  of package diagrams, 266–268
  of physical architecture layer design, 449
  of structural modeling, 194–197
Version 2.5 (UML), 34, 35
Virtualization, 426
Virtual memory, 426
Virtual reality (VR), 404
Virus control, 477
Virus control requirements, 447
defined, 445
Visibility
  of attribute, 176
  of methods, 282
  schedule, 17
Visualization, 147
Volume tests, 477
Volumetrics, 356

W
Walkthrough, 108
evaluation, 381
verification and validation through, 154–155
Waterfall development, 7
WBS, see Work breakdown structure
Web services, 270, 426
White-box testing, 472, 473
White space, 370
Wholes, 166
Whole-system conversion, 487
Windows layout diagram, 372, 377
Windows navigation diagram (WND), 372, 375
Workaround, 270
Work breakdown structure (WBS), 55–56
  evolutionary, 63–67
Workflow modifications and extensions
  configuration and change management workflow, 33–34
  deployment workflow, 32
  environment workflow, 32
  project management workflow, 33
  test workflow, 32
Workflows
  engineering, 28–29
  supporting, 29–30
  in Unified Process, 28–30
Workplan creation and management, 63–71. See also Project management
  estimates, refining, 69–70
  evolutionary work breakdown structures, 63–67
  iterative workplans, 63–67
  risk management, 70–71
  scope management, 67–68
  timeboxing, 68–69

X
XP, see Extreme programming

Y
Yourdon, Edward, 154, 286, 289
Yo-yo problem, 476