

Index

- Adapters. *See also* Data adaptation
 - use in Retail Supply System (RSS), 237–239
- Aggregate Entity pattern, data components, 201, 202
- Aggregation hierarchies, 64
 - Apache Xerces, 110
- Applets, 98. *See also* Java
- Application components, 126
 - elements of, 198
 - structure of, 198
- Application Programming Interfaces (APIs), 97, 98–99. *See also* Java
- Architectural compliance, 44
- Architectural patterns
 - ad hoc query, 211
 - data access
 - with more than one business object, 207, 208
 - within one business object, 206, 207
 - data warehouses, 215–219
 - report, 209–211
 - roll-ups, 211–212
 - transactions, 215–216
- Architectural views
 - component-and-connector (C&C), 76–83
 - decomposition, 74
 - deployment, 83–84
 - generalization, 74–75
 - hybrid, 85
 - layered, 76
 - module, 73–76
 - system context, 84–85
- Architecture reconstruction, 32
 - definition of, 64–65
 - phases of, 65–66
- Architecture representation, 64
 - as-built, 69
 - as-desired, 69, 70
 - levels of granularity, 72
 - purpose of, 71
 - requirements of, 71–72
- Architecture transformation, 60, 221
 - adapters used in, 237–239
 - code migration. *See* Code migration
 - data migration. *See* Data migration
 - deployment options during, 235–236
 - operational risk in, 235
 - strategies for, 229–236
 - trail maps used in, 236–237
- Artifact extraction, 61
- Asynchronous communication, 114, 115
- Automatic translation, 13, 14

- Batch roll-ups, 212–213
- B2B architecture, 173–174
 - availability in, 182
 - data integrity of, 183
 - evaluation of, 179
 - integrability of, 188
 - versus* J2EE architecture, 176, 190
 - modifiability of, 184
 - performance in, 177
 - portability of, 187
 - reusability of, 187
 - security of, 180
- Bean-managed persistence (BMP), 137, 138
- BEA Tuxedo, 131–132
- BEA WebLogic (EJB Server), 118
- Behavioral pattern matching, 63
- BizTalk
 - framework, 150
 - and XML messaging, 149–150

324 Index

- Black-box modernization, 9
- BSR (business service request)
 - interaction sequence for, 218
 - processing of, 205–206
- BSR constructor, 203
- BSR interface
 - elements of, 203, 204
 - layered architecture of, 203
- BSR interpreter, 203
- Business case
 - contents of, 46–47
 - cost-benefit analysis in, 47, 53–54
 - creation of, 39–40
 - problem statement, 50
 - quantifiable benefit metrics, 48–49
 - risk identification, 53
- Business logic integration
 - adaptation techniques, 228, 229
 - and use of connectors, 229
- Business objects
 - and application components, 199
 - of RSS, 195, 196
 - size of, 264
- Business process reengineering (BPR),
 - requirements generated by, 42
- Business service request. *See* BSR
- Business value, as portfolio analysis
 - measure, 29
- Call graph, for program element migration, 254–255
- COBOL
 - arithmetic of, 91
 - call statements, 92
 - conversion of, 13, 293–294
 - data items, 91
 - features, 92–93
 - history of, 88–89
 - interfacing with Java, 166–168, 293–294
 - products, 94
 - program structure, 89–90
 - standards, 93
 - variable declarations, 91–92
 - worldwide use of, 5
- Code instrumentation, 63
- Code migration, 230–232
 - alternative plans for, 259–264
 - and data migration, 233–234, 264–265
 - prioritization results, 285–287
 - by program element sets, 231
 - replacement with commercial components, 12
 - and related functionality, 232
 - and user transactions, 231
- Code reduction, 14–15
- Code-structure representation, 61–63
 - artifact extraction, 61
 - dynamic analysis, 62–63
 - manual code reading, 61
 - static analysis, 61–62
- Code transformation, 59–60
- Collaboration diagrams, 82–83
- Collaboration Protocol Agreement (CPA), 151–152
- Collaboration Protocol Profile (CPP), 151–152
- Commercial components
 - functional, 13
 - infrastructure, 13
- Commercial off-the-shelf (COTS) products, 44, 304
- Commit point, 120
- Communication resource manager, 126
- Communication technologies
 - asynchronous, 114, 115
 - connection-oriented, 114, 115, 116
 - direct, 114, 115
 - distributed, 114–119
 - products, 118
 - transactions model, 113, 119
 - message-passing, 116, 117
 - message-queuing, 116, 117
 - publish/subscribe, 116
 - queued-transactions model, 113, 119
 - RPC-based, 113
 - synchronous, 114, 115
- Compensating transactions, 179
- Component-and-connector views, 76–83
 - Notations, 78–83
 - collaboration diagrams, 82–83

- sequence diagrams, 80–81
 - use case maps, 79–80
- Component-based development, 306
- Component ensembles, 34
- Componentization. *See* Architectural transformation
- Component wrapping, 227–228
- Concept assignment, 64
- Concurrent development, 307
- Conference on Data Systems Languages (CODASYL), 88, 104–105
- Connection-oriented communication, 114, 115, 116
- Constructive cost model II (COCOMO II), 297
- Container-managed persistence (CMP), 137, 138
- Content handler interface, 110
- Context diagram, 85
- Contingency planning, 160–163
- Continuous integration, 307
- Continuously updated roll-ups, 214–215
- Cost avoidance, 47
- Cost estimation
 - function-based, 290–292
 - models. *See* cost models
 - task-based, 292
- Cost models
 - of effort *versus* duration, 296–297
 - need for calibration of, 297
- Cost reduction, 47
- Data access
 - layer, 224, 226
 - patterns, 206–208
- Data adaptation, 222–226
 - evaluation of techniques, 225–226
 - generic, 222
 - hybrid, 225
- Database gateway, 224–225, 226
- Database migration, cost estimation of, 299
- Databases
 - distributed, 193
 - hierarchical, 100
 - network, 100–101
 - object-oriented, 102–103
 - relational, 101, 102
 - transactional update of, 159
- Database standards, 104–106
- Database triggers, 224, 226
- Data collection for cost estimation, 299–300
- Data components
 - bean-managed persistence, 200
 - common qualities of, 202
 - container-managed persistence, 200
 - internals of, 202
 - use of Aggregate Entity pattern, 201
- Data Management System (DMS) 2200, 106
- Data marts, 104
- Data migration, 232–234
 - after code migration, 234, 264
 - before code migration, 233
 - during code migration, 233–234
 - plan for, 264–267
- Data reengineering, 15
- Data replication, 222–224
- Data representations, for information exchange, 108–111
- Data type definition (DTD), 203
- Data warehouse population
 - interaction sequences for, 218–219
 - use of pull mechanism for, 216–217
 - use of push mechanism for, 217, 219
- Data warehousing, 104, 193, 215–219
- DBMS (database management system), 100–103
 - products, 106–107
- Decompilation, of source code, 66–67
- Decomposition view, 74, 75
- Dependent data marts, 104
- Deployment view, 83–84
- Development and deployment
 - incremental, 16, 44–45
 - minimizing costs of, 45
- Development tasks, cost estimation of, 292
- Direct communication technologies, 114, 115
- Disassembly, of source code, 66–67
- Distributed communication. *See* Communication technologies, distributed

326 Index

- Distributed databases, 193
- Distributed-transaction model, 113, 119, 124–126
 - evaluation of, 129
- Distributed transactions, 124–129
 - EJB support of, 139
- Document object model (DOM), 109, 110
- Dynamic analysis techniques, 62–63
- Dynamic pages, 24

- ebXML. *See* Electronic Business XML (ebXML)
- EDI (electronic data interchange), 108
 - industry acceptance of, 108
- EJB server, used in SRF, 194
- Electronic Business XML (ebXML), 150–152
 - architecture for, 151
 - usage scenario for, 151
- Enterprise JavaBeans (EJB), 98, 136–140
 - architecture, 137
 - entity, 137, 138, 201, 265
 - message-driven, 137, 138
 - products, 140
 - security management for, 139–140
 - session, 137–138, 199–200
 - as support for distributed transactions, 139
 - types of, 137–138
- Entity beans, 137, 138, 201
 - as persistence layer, 138, 265
- eXtensible Style Language (XSL), 109
- Extraction, transformation, and loading (ETL), 223, 226
- Evolvable software development, 6

- Fast-Lane reader pattern, 209
- Functional components, 13
- Functionality, measures of, 291
- Functional transformation, 15–16
 - program structure improvement, 15
 - program modularization, 15
 - data reengineering, 16
- Function-level representation, 63–64
- Function-point metric, 291
- Fused view, 65

- Generalization view, 74–75
- Generic query builder, 198
- Graphical user interface (GUI), 198

- Hierarchical databases, 100
- Horseshoe model, 58, 59
- Host platform factors, 297
- HTML, use in SRF, 194
- Hybrid view, 85

- IBM
 - MQSeries. *See* MQSeries (IBM)
 - Transarc Encina, 132
 - XML4J, 111
- Incremental deployment, 305
- Incremental development, 305
- Incremental life cycle, 293
- Increments
 - cost estimation of, 292–293, 297–298
 - size estimating of, 292–295
 - testing of, 297–298
- Information exchange, data representations for, 108–111
- Information system technology, classes of, 33
- Infrastructure components, 13
- Integration, continuous, 307
- International COBOL Committee, 89

- Java
 - Application Programming Interfaces (APIs), 97, 98–99
 - conversion to, 14–15, 293–294
 - history of, 94
 - interfacing with COBOL, 166–168, 293–294
 - language characteristics, 97–98
 - platform, 95–97
 - products, 99
 - programming language, 194
 - program structure, 95
 - program types, 98
 - servlets, 194
- JavaBeans, 98
- Java Database Connectivity (JDBC), 105–106

- Java 2 Enterprise Edition (J2EE), 146–148
 - architecture of, 146, 147
 - deployment of, 148
 - services, 148
- Java Message Service (JMS), 118, 138
- Java Naming Directory Interface (INDI), 165
- Java Native Interface, 98
- Java runtime environment (JRE), 97
- Java Server Pages (JSPs), 194
- Java Transaction API (JTA), 131, 165
- Java Transaction Service (JTS), 131
- Java Virtual Machine (JVM), 97
- JBoss, 140
- JDBC-ODBC bridge, 225
- J2EE architecture, 173
 - Aggregate Entity pattern, 201, 202
 - availability in, 182
 - versus* B2B architecture, 176, 190
 - data integrity of, 183
 - evaluation of, 179
 - Fast-Lane reader pattern, 209
 - integrability of, 188
 - modifiability of, 186
 - order placement scenario in, 173, 174
 - performance in, 178–179
 - portability of, 187
 - reusability of, 187
 - security of, 181
 - Session Entity Façade pattern, 199
- Labor, cost model, 296
- Layered view, 76
- Legacy crisis
 - counterforces of, 6–7
 - definition of, 6
- Legacy system
 - adapters, 237–239
 - after code migration, 265, 266
 - after data migration, 265, 267
 - architecture-centric development of, 306–307
 - before code migration, 265
 - during code migration, 265, 266
 - component-based development of, 306
 - life cycle of, 7, 8
 - logic wrapping of, 226–228
 - maintenance of, 8, 9
 - modernization of. *See* Modernization
 - preparation of, 35
 - evaluation of plans, 252
 - migration of data alternatives, 242–251
 - migration to Solaris platform, 250–251
 - reengineering of, 10–11
 - replacement of, 10
 - requirements generated by, 42
 - retargeting of, 11
 - revamping of, 11–12
 - screen scraping technique for, 11
 - size estimation, 293–294
 - structural analysis of, 254–259
 - understanding architecture of, 31–32
- Legacy system modernization. *See* Modernization; Risk-managed modernization
- Lehman's Laws
 - first law, 4
 - second law, 5
- Logic adaptation techniques
 - component wrapping, 227–228
 - evaluation of, 228–229
 - object-oriented wrapping, 226–227
- Maintenance. *See* Software maintenance
- Manual code reading, 61
- Message descriptor, 142
- Message-driven beans (MDBs), 137, 138
- Message-oriented middleware (MOM), 140–142
- MessageQ, 118
- Message queues, 141, 143–144
 - attributes of, 143
 - channel communication, 144
 - remote, 144
 - technology of, 116, 119
- Migration. *See* Code migration; Data migration
- Migration trail maps. *See* Trail maps

328 Index

- Model problem
 - architecture for, 165
 - definition of, 163–164
 - design of, 158–160
 - ending evaluation of, 169
 - solution of, 164–165
- Modernization. *See also* Architectural transformation; Risk-managed modernization
 - adapters used in, 237–239
 - alternate approaches to, 243–251
 - benefits of, 48, 53–54
 - complexity of, 304
 - contingency planning in, 160–163
 - cost of, 52
 - definition of, 9
 - failure of, 1–3
 - feasible plan for, 289
 - goals of, 45–46
 - and horseshoe model, 58, 59
 - incremental, 47
 - of operational system, 159
 - processes of, 58
 - reconciled with stakeholder needs, 35–36, 270
 - versus* replacement, 48
 - requirements
 - and architecture compliance, 44
 - constraints on, 43
 - sources of, 42
 - risks of, 53
 - schedule, 51–52
 - strategy of, 35
- Module views, 73–76
 - decomposition, 74, 75
 - generalization, 74–75
 - layered, 76
 - UML notations in, 73
 - UML relations in, 73, 74
- MQSeries (IBM), 118
 - architecture of, 141
 - for ClearPath OS 2200, 146
 - as design contingency, 160–161
 - interface, 145
 - message format of, 142
 - message types, 142
 - messaging styles, 143
 - products, 146
 - retrieval options, 142–143
 - transaction support for, 145
 - triggers, 145
 - used in SRF, 194
- Net Express, as mixed-language programming support, 162–163
- Network databases, 100–101
- Node, definition of, 84
- Node program elements, 254
- Nonfunctional requirements, 30
- Nonparallel operations, as deployment strategy, 236
- OAGIS, 152, 153
 - evaluation of, 184
 - release history of, 185–186
 - used in SRF, 194
- OAMAS, 153
- Object Data Management Group (ODMG), 106
- Object-oriented databases, 102–103
- Object-oriented wrapping, 226–227
- Object Query Language (OQL), 106
- Object Request Broker (ORB), 99
- Object Transaction Service (OTS), as design contingency, 161
- OCL (Object Constraint Language), 81
- On-line analytical processing (OLAP), 99
- On-line transaction processing (OLTP), 99
- On-the-fly roll-ups, 212
- Open Applications Group Integration Specification. *See* OAGIS
- Open Applications Group (OAG), 152–153
 - business-object model of, 152
 - mission of, 152
- Open Applications Middleware API Specification. *See* OAMAS
- Open Database Connectivity (ODBC), 105
- Oracle, 106
 - migration to, 248–250, 265
 - on OS 2200 environment, 248–250
 - transfer of data to, 25–26
 - used in SRF, 194
- Oracle Discoverer, 107

- Oracle Pro*COBOL, as design contingency, 161–162
- OS 2200 environment
 - DMS on, 244
 - migration of data from, 243–246
 - Oracle on, 248–250
 - RDMS on, 246–247
- OTS (Object Transaction Service), 131
- Parallel operations, as deployment strategy, 235–236
- Partner interface processes (PIPs), 154
- Pattern definition, 66
- Pattern recognition, 66
- Persistent roll-ups, 212
- Personnel factors, 297
- Plan recognition, 63
- Portfolio analysis, 29–30
- Preparation work, cost estimation of, 298
- Productivity ratio, in cost/duration estimation, 291, 295–296
- Profiling, 62
- Program comprehension, 9
 - techniques of, 61–64
- Program elements
 - kinds of, 254–255
 - migration plan, 256–258, 258–259
 - reachable, 257
 - sample documentation of, 258
- Program modularization, 15
- Program slicing, 63
- Program structure improvement, 15

- Quality attributes
 - availability, 181–182
 - categories of, 176
 - data integrity, 183
 - integrability, 188–189
 - modifiability, 184–186
 - performance, 176–179
 - portability, 186–187
 - reliability, 187–188
 - reusability, 187–188
 - security, 180–181
- Queued-transactions model, 113, 119, 127–129
 - evaluation of, 130
- Queue manager, 144

- Rational Rose Extensibility Interface (REI), 256
- RDMS (Relational Data Management System) 2200, 107
- Reader interface, 110
- Reconciliation
 - objectives of, 270
 - plan, 271
- Reconstruction process, 58, 59, 61–64
 - aggregation hierarchies, 64
 - architecture and structure identification, 64
 - artifact extraction, 61
 - behavioral pattern matching, 63
 - concept assignment and reasoning, 64
 - dynamic analysis, 62–63
 - manual code reading, 61
 - plan recognition, 63
 - refactoring, 64
 - slicing, 63
 - semantic pattern matching, 63
 - static analysis, 61–62
 - structural pattern matching, 64
 - redocumentation, 63
- Redocumentation, 63
- Reengineering.
 - of architecture. *See* Architectural transformations
 - of data, 15
 - manual *versus* tool-supported, 66
 - of software, 6, 10, 11–15
- Refactoring, 64
- Refinement process, 58, 59
- Relational databases, 101, 102
- Remote Method Invocation (Java API), 99
- Remote queues, 144
- RENNAISSANCE project, 48
- Replacement
 - "big-bang" approach, 10
 - risks of, 10
- Report pattern, 209–211
- Resource manager, 125
- Retail Supply System (RSS)
 - application, 196
 - architecture of, 171, 196, 197
 - business case for, 49–54
 - business objects of, 195, 196

330 Index

- Retail Supply System (RSS) *continued*
 - characteristics of, 20–21
 - collaboration diagram for, 82
 - data and control flow in, 22
 - data requirements of, 192–193
 - deployment diagram for, 83
 - design solutions for, 158–159
 - distributed organization of, 193
 - environmental factors, 50
 - functionality of, 43
 - high-level architecture of, 21
 - incremental development strategy, 293
 - management reports provided by, 25
 - modernization cost of, 52
 - modernization goals of, 45–46
 - modernization schedule for, 51
 - quantifiable benefit metrics, 50
 - sequence diagram for, 81
 - SRF architecture for, 171
 - stakeholders of, 40–41
 - use case map for, 80
 - Web enablement of, 23–25
- Retargeting, 11
- Revamping, 11–12
- Risk identification
 - in business case, 47
 - during early stages, 53
- Risk-managed modernization (RMM), 27, 28, 308. *See also* Modernization
 - business case for, 31
 - cost of, 36
 - evaluation of existing technologies, 32–34
 - general application of, 308
 - requirements of, 30–31
 - and stakeholders, 30, 35–36
 - strategy for, 35
 - and target architecture, 34
 - UML activity diagram of, 28
 - and understanding the legacy system, 31–32
- Risk reduction, 308. *See also* Risk-managed modernization (RMM)
- Roll-ups, 211–212
 - batch, 212–213
 - continuously updated, 214–215
 - on-the-fly, 212
 - persistent, 212
- Roll-up table, 214
- RosettaNet
 - components offered by, 154
 - membership of, 153
- Save point, 120
- SAX (Simple API for XML), 109, 110
- Schedule compression, 292
- Screen scraping, 11–12
- Scripts, 223, 225
- Semantic pattern matching, 63
- Sequence diagrams, 80–81
- Service broker, 227
- Service components, 199–200
- Servlets, 98. *See also* Java
- Session beans, 137–138, 199–200
- Session Entity Façade pattern, 199
- Session manager, 198
- Simple Object Access Protocol (SOAP), 155
- Snooping, 63
- Software engineering, 305
- Software maintenance
 - adaptive, 4
 - categories of, 4
 - corrective, 4
 - cost estimate of, 294–295
 - cost of, 5, 6
 - definition of, 8
 - difficulty of, 5
 - perfective, 4
 - preventive, 4
- Software product factors, 297
- Software project factors, 297
- Software requirements, 30–31
 - constraints on, 30
 - criteria for, 31
- Software Risk Evaluation (SRE) method, 53
- Software technologies
 - classes of, 33
 - evaluation of, 34
- Solaris platform
 - migration of code base to, 250–251
 - oracle on, 248–249

- Source code
 - expansion of, 3–4
 - recovery of, 66–67
 - translation, 13
- SQL (Structured Query Language), 105
- SRF (Standard Retail Framework), 44
 - B2B architecture, 173
 - and business object integration, 171
 - J2EE architecture, 173
 - technologies contained in, 194
- Staffing, *versus* duration of a project, 307
- Stakeholders
 - architecture team as, 272–273
 - consensus meeting of, 284
 - and consensus profile, 285–287
 - ideal profiles of, 275
 - identification of, 30, 40–41
 - legacy system maintainers as, 273–274
 - management as, 274
 - priorities of, 271–275
 - reconciliation with, 270
 - requirements generated by, 42
 - user representatives as, 40, 271–272
- Standard Retail Framework. *See* SRF (Standard Retail Framework)
- Stateful session beans, 200
- Stateless session beans, 199–200
- Static analysis techniques, 61–62
- Static pages, 24
- Structural pattern matching, 64
- Sun JAXP, 111
- Synchronous communication, 114, 115
- System qualities. *See* Quality attributes

- Target architecture
 - data requirements of, 192–193
 - evaluation of, 34
 - organizational requirements of, 193
 - technology requirements of, 193–194
- Technical quality, as portfolio analysis measure, 29
- Technology, advances in, requirements generated by, 42
- Total asset visibility, 43
- Trail maps, 236–237
- Transactional models
 - comparison of, 129–131
 - distributed-transaction, 124–126, 129
 - queued-transaction, 127–129
- Transaction context, 123, 124
- Transaction manager, 125, 126, 129
- Transaction processing (TIP), 23
 - mechanisms, 24
- Transactions
 - chained, 120, 121
 - compensating, 122, 129
 - distributed-access, 123, 124
 - flat, 120, 121
 - global, 124, 129
 - multilevel, 123
 - nested, 121–123
 - closed, 121–122
 - open, 122–123
 - simulated, 122, 123
 - properties of, 119
 - with save points, 120, 121
 - sequence diagram for, 215–216
 - technologies of, 131–132
- Transaction technologies
 - products for, 131–132
 - standards for, 131
- Transformation, 58, 59
 - architectural, 59, 60
 - code, 59–60
 - functional, 59, 60
 - levels of, 59–60
- Trigger monitor, 145
- UML (Unified Modeling Language), 70
 - class concept in, 78
 - deployment diagram in, 84
 - generalization in, 75
 - layer representation in, 76
 - module notations, 73, 74
 - Object Constraint Language (OCL), 81
 - relation notations, 74
- UniAccess for OS 2200, 107
- Unisys
 - Data Management System (DMS 2200), 20
 - database storage structures, 21–22
 - transaction processing, 23
 - Display Processing System (DPS 2200), 23
- Unisys Data Access, 107

332 Index

- Universal Data Management System (UDS 2200), 107
- Universal Description, Discovery, and Integration (UDDI), 155
- Use case maps, 79–80
- User interface (UI), revamping of, 11
- User requirements, 30

- View extraction, 65
- VisiBroker (CORBA), 118
- Visualization and interaction, 65–66

- Web enablement, of RSS, 23–25
- WebLogic, 140
- Web Services Description Language (WSDL), 155
- WebSphere, 140
- WebSphere MQ, 146
- Web Transaction Server (WebTS), 24
- White-box modernization, 9
- World Wide Web Consortium, development of XML, 108
- Wrapper components, 202
 - interaction sequence for, 219–220
 - types of, 203

- Wrapping
 - definition of, 9
 - screen scraping technique, 11–12

- XML, 108–110.
 - DOM (document object model), 109, 110
 - DTD (document type definition), 109
 - versus* HTML, 109
 - messaging. *See* XML messaging
 - parsers, 109, 203
 - SAX, 109

- XML messaging, 148–149
 - and BizTalk, 149–150
 - components of, 149
 - and ebXML, 150–151
 - and OAGIS, 152, 153
 - and RosettaNet, 153–154
 - and SOAP, 155
 - and UDDI, 155
 - and WSDL, 155

- XML schemas, 109, 110, 203
- X/Open DTP, 132
 - reference architecture, 124