

# INDEX

One of the great defects of English books printed in the last century is the want of an index.

—Lafcadio Hearn a.k.a Koizumi Yakumo

- Abstract syntax tree (AST), 157, 318, 358
- Abstraction, 135–136, 142, 149
- Accelerated development, 327
- Acceptance
  - test, 98–99
  - testing, 153
- Accidental reuse, 358
- Accounting, 116
- Activity, 40, 358
- Activity definition, 95
- Activity-based classification, 28
- Actual impact set (AIS), 225–226, 358
- Ad hoc* reuse, 338
- Ada SDA (System Dependency Analyzer), 171
- Adaptability, 16, 326
- Adaptive, 1, 30–31, 289
- Adaptive maintenance, 27, 358
- Adequacy, 227, 359
- Adherence, 227–228, 274, 359
- Age, 39
- Agglomerative algorithms, 281
- Agile
  - programming, 58
  - software development, 359
  - software methodology, 13
- Aging, 53, 375
- Aiken, Peter, 11
- Albrecht, A. J., 162
- Alteration, 10, 137–138, 140
- Amplification, 227
- Analysis
  - knowledge, 317
  - and planning, 150–151
  - of protocol data, 315–316
  - state, 123
- Annotation, 164, 300
- Annotation of the program, 359

- Anomaly, 60
- Antiregression, 35, 58, 359
- Antithesis, 54
- Application
  - domain, 39
  - engineering (AE), 330–331, 348, 359
  - gateway, 206, 359
  - program interface (API), 189
  - wrappers, 191
- Approval, 104
- Approval task steps, 106
- Architectural reuse, 15, 326
- Architecture description languages (ADLs), 330
- Architecture extraction, 135
- Arnold, Robert S., 176
- Arthur, Jay Lowell, 2, 45
- Artifacts, 12, 38, 42, 133, 258, 359
- Aspect, 360
- Aspect-oriented software development, 155, 360
- Assembler, 165, 360
- Assertion, 164, 265, 360
- Assessment model, 336
- Asset, 360
- Assimilation process, 15, 301
- Association coefficient, 280
- Assumptions management, 56
- Audit, 116
- Auditing, 116, 118
- Automated analysis, 257
- Available support community, 68
- Backus–Naur Form (BNF), 157
- Backward
  - search, 235
  - slice, 159
  - wrappers (b-wrappers), 189, 191, 360
- Bad smell, 360
- Baseline, 37, 118, 360
- Basic conceptualization, 170
- Basic reuse, 338–339
- Bazaar, 58
- Beacon, 296, 360
- Behavior preserving transformation, 266
- Belady, Laszlo, 1
- Bennett, Keith H., 45
- Behavioral dependencies, 69
- Big bang, 360
- Big Bang approach, 144, 202
- Black-box
  - components, 5
  - reengineering, 11, 189
  - requirements, 346
  - testing, 360
- Bottom-up, 301
- Branch, 114
- Brooks model, 303
- Build process, 70
- Building, 115
- Business model of Poulin and Caruso, 350
- Business
  - process reengineering (BPR), 199
  - rules, 159
- Butterfly, 12, 360
- Butterfly methodology, 208
- Byrne, Eric J., 138
- Call graph, 13, 158, 234–235, 317, 360
- Call graph-based analysis, 13, 224
- Call preservation, 262
- Call, use and define (CUD), 245
- Candidate impact set (CIS), 225–226, 231, 360
- Canning R. G., 1, 25
- Capability Maturity Model (CMM), 64
- Carry on maintenance, 11, 188
- Cathedral, 58
- Causation, 149
- Centroid, 281
- Certification, 149, 329
- Change
  - control, 42, 361
  - coupling, 361
  - management, 115, 361
  - mini-cycle model, 7, 91
  - propagation, 13, 245, 265
  - propagation activity, 94, 225, 361
  - propagation heuristics, 243, 247
  - propagation model, 242–243
  - rate, 228
  - request (CR), 26, 43, 92, 116, 119, 215, 223, 225, 243
  - request schema, 121
  - set, 243–244
  - source instruction (CSI), 350
  - system, 42

- Chapin, Ned, 28–29, 45
- Chicken little, 12, 361
- Chicken little
  - migration, 207
  - strategy, 206
- Chidamber and Kemerer (CK) metrics, 162, 164
- Chikofsky and Cross II, 153–154
- Chrysaliser, 208, 210–212
- Chunk, 293, 361
- Chunking, 15, 295, 361
- City metaphor, 320
- CK metric suite, 162
- Class diagram, 260, 262, 265–266
- Cleansing, 195
- Clearcase, 115, 118
- Cliché, 171, 361
- Client organization, 43
- Clone, 61, 164, 177, 375
- Closed source software (CSS), 3, 49, 58, 87, 361
- Closed state, 125
- Closedown, 88–89
- Clustering, 278, 280
- Clusters
  - business rules, 29, 31, 36
  - documentation, 29, 31, 34
  - software properties, 29, 31, 35
  - support interface, 29, 31, 34
- COCOMO model, 199
- Code
  - changes, 275–276
  - churn, 361
  - cognition, 291
  - decay, 53–54, 361
  - layout (FIL), 246
  - leverage, 289
  - readers, 297
  - smell, 258, 361
- CodeCrawler, 171
- Cognitions, 317
- Cohesion, 263
- Cohesion metric, 267–268
- Cold turkey, 12, 201, 361
- Commercial off-the-shelf (COTS), 5–6, 36, 62, 361
- Commit state, 123
- Common goals, 44
- Compilation, 141, 152, 165, 361
- Complex refactoring, 14
- Complexity, 263
- Complexity metrics, 162
- Compliance, 327
- Component
  - collaboration, 68
  - reconfiguration, 65
  - reuse, 167
  - selection, 67
- Component-based software (CBS), 5–6, 62
- Composite database, 12, 362
- Composite database approach, 205
- Comprehension strategies, 308
- Computationally redundant data, 213
- Computer-aided software engineering (CASE), 40
- Concept assignment problem, 228
- Concepts, 93, 228, 362
- Conceptual
  - data, 213
  - level, 136
  - normalization, 170
  - phase, 168
  - schema, 167, 195, 362
- Concurrent version system (CVS), 118
- Configuration, 362
  - baseline, 117
  - control board (CCB), 118
  - item, 362
  - management (CM), 9, 42, 66–67, 103, 111, 148, 362
- Conformance, 150
- Conjectures, 14, 295, 299
- Connectivity matrix, 232
- Conservation of familiarity, 3, 39, 50, 52, 60, 242
- Conservation of organizational stability, 3, 50, 52, 60, 242
- Construction, 362
- Construction algorithms, 280
- Constructive Cost Model II (COCOMO II), 202
- Consultive, 30–31
- Context area, 318
- Continuing change, 3, 50–51, 134, 242
- Continuing growth, 3, 50, 52, 134, 242
- Contours, 320
- Contracting, 12

- Control
  - data, 213
  - dependency edges, 236
  - flow, 292, 312
- Control flow
  - analysis (CFA), 10, 362
  - graph (CFG), 142, 158, 317
- Control structure diagram (CSD), 318
- Controlled interfaces, 69
- Controlling, 118
- Corrections, 28
- Corrective, 1, 30–31, 289
- Corrective maintenance, 26, 362
- Correlation coefficients, 280
- Cost estimation, 12
- Cost model of Gaffney and Durek, 340
- Cost per error, 350
- Cost-benefit analysis, 12, 200
- Coupling, 54, 263
- Coupling between object class (CBO), 163
- Coupling metrics, 268
- Critical pair analysis, 261
- Crosscutting concerns, 362
- Cross-referencing, 15, 295–296, 362
- Customer, 362
- Cut over, 196, 362
- Cut-and-run, 196
- Cyclomatic complexity, 162–163, 197–198, 241
- Dangling purpose, 301
- Data
  - access-allocator (DAA), 208, 210–212
  - administration, 166, 362
  - analysis, 169
  - archival, 109, 111
  - assets, 166
  - banker, 214
  - cleaning, 195
  - conversion, 167, 195, 362
  - dependency edges, 235–236
  - description language (DDL), 168
  - flow, 292, 312
  - flow analysis (DFA), 10, 158, 363
  - flow graph (DFG), 317
  - forward engineering, 166
  - locator, 214
  - management system (DMS), 168
  - reduction, 148
  - reengineering, 152
  - representation, 148
  - reuse, 15, 326
  - reverse engineering (DRE), 11, 166, 363
  - structure conceptualization, 168
  - structure extraction, 168
- Database, 166
- Database
  - first, 12, 189, 203
  - gateway, 206, 362
  - last, 12, 204
  - management system (DBMS), 167–168
  - reverse engineering (DBRE), 167, 195, 363
  - wrappers, 189
- Data-oriented application, 165–167
- Data-Transformer, 208, 212
- Davis, Ted, 336
- Decay, 363
- Decision tree, 32
- Decision tree-based criteria, 31
- Declining quality, 3, 50, 53, 134, 242
- Decompilation, 135, 141, 165, 363
- Decompiler, 152, 164
- Definition-use, 317
- Degree of interest (DOI), 318
- Delivery, 99
- Density search technique, 280–281
- Dependency analysis, 12
- Dependency graph, 13, 229, 234
- Dependency graph-based analysis, 13, 234
- Depth of inheritance tree (DIT), 163
- Design, 96
- Design
  - level, 136
  - recovery, 154, 363
  - reuse, 16, 326
  - clones, 164
- Developer, 2
- Developer information (DEV), 246
- Development
  - cost avoidance, 351
  - model, 363
  - process, 111, 363
  - replay (DR), 246
  - for reuse, 325, 328, 329
  - with reuse, 325, 330
  - staff, 44
  - technology, 41, 363

- Device-drivers, 61
- Direct impact, 231
- Disassembler, 164, 363
- Discard and redevelop, 11, 188
- Discovered impact set (DIS), 225–226, 363
- Distance measure, 280
- Distance-based, 268
- Distance-based approach, 233
- Distinctive appearance, 160
- Divide-and-conquer schema, 297–298
- Divisive algorithms, 281
- DMS (Design Maintenance System), 171
- DMS–DDL text analysis, 169
- Documentation, 104, 146, 166, 274
  - change, 94
  - task steps, 105
- Documenting, 118
- Domain, 325, 363
  - analysis, 15, 325, 329, 36
  - design, 330, 364
  - engineering (DE), 328–329, 331, 348, 364
  - implementation, 330, 364
  - knowledge, 303, 304, 308, 314
  - model, 313, 331, 364
  - plans, 294
  - qualification (DQ), 331
  - specific languages (DSLs), 364
- Domain analysis and reuse environment (DARE), 331
- Draco, 331
- Duplicate code, 259
- Dynamic
  - call graph, 158
  - elements, 295
  - growth, 44
  - program slice, 225, 237–238
- Economic, 328
- Effective size, 346
- Effective visual metaphors, 162
- Effectiveness, 227, 363
- Eighth law, 50, 53
- Element connectedness, 320
- Elimination-of-goto, 276
- Embedded knowledge, 53
- Empirical studies, 54, 60
- Encapsulation, 146–148
- Encoding, 316
- Engineering change (EC), 112, 124
- Engineering view point, 46
- Enhancements, 28
- Enhancive, 30–31
- Enterprise JavaBeans (EJB), 201
- Entity-relationship (ER), 167
- Entropy, 53, 375
- Environment, 275
- Environmental independence, 326
- Equivalence tests, 217, 364
- Error flow analysis, 238
- Establishing baseline, 118
- Estimated impact set (EIS), 93
- E-type program, 48, 51, 134, 364
- E-type software, 56
- E-type systems, 3
- Euclidean distance, 268
- Evaluation, 317
- Evaluative, 30–31
- Event management, 42, 364
- Evidence-based classification, 28
- Evolution, 1, 7, 44–45, 87–88
- Evolution
  - of E-type software, 49
  - management, 57
  - process, 49, 51, 53, 89, 364
  - of software, 44
- Evolutionary
  - approach, 146
  - model, 92
  - trajectory, 51
- Evolves, 2
- Exception handling, 292
- Execution history, 237
- Explanatory (what/why), 45
- Extensibility, 256, 263
- External
  - documentation, 307
  - interface, 192
  - representation, 300
  - reuse (a.k.a. public reuse), 364
  - software value, 256
- External traceability, *see* Horizontal traceability
- Extract, transform, and load (ETL), 195
- Extractive, 327–328
- Extreme programming (XP), 13, 365

- Facebook, 330
- Failure handling, 70
- False positive, 233
- False positive impact set (FPIS), 225–226, 365
- Familiarity, 320
- Fan-in, 162–163
- Fan-out, 162–163
- Family-oriented abstraction, specification, and translation (FAST), 331
- Feature location, 229
- Feature-oriented domain analysis (FODA), 330
- Feature-oriented reuse method (FORM), 331
- Features, 228
- Feedback, 365
- Feedback system, 3, 49, 50, 53, 242
- Feedback, evolution, and software technology (FEAST), 54
- FermaT, 172
- Fifth law, 50, 52
- First law, 50–51
- First order reuse cost model, 346
- Fisheye view, 318
- Focus area, 318
- Forward
  - engineering, 10, 135, 139, 154, 365
  - gateway, 203, 205–206, 365
  - migration, 203, 365
  - searching, 235
  - slicing, 159, 225
  - software engineering principles, 210
  - wrappers (f-wrappers), 189, 365
- FOSS evolution, 60
- Fourth law, 50, 52
- Framework, 365
- Free and Open Source Software (FOSS), 4, 58, 365
- Free software, 4, 58
- Free software, 58
- Free/Libre/Open Source Software (FLOSS), 58, 90
- Freeze, 11, 188
- Frequency, 246
- Friendly user interface, 162
- Frozen, 209
- Frozen functionality, 63
- Full reuse model, 85, 87
- Fully decomposable, 204–205, 365
- Function, 146, 312
- Function
  - name, 297
  - point (FP), 162, 163
  - wrappers, 191
- Functional
  - knowledge, 314–315
  - requirements, 256
  - enhancement, 134
- Functionality, 292
- Functionalization, 149
- Gate keepers, 59
- Gateway, 203, 365
- Gateway co-ordinators, 208
- General knowledge, 291
- Generation of
  - code, 149
  - design, 149
- Generic knowledge structure, 294
- Geographic technique, 280–281
- Gestalt's theory, 319
- Gibson's theory, 319
- Global
  - factors, 60
  - variable (GV), 163
- Glue, 62–63, 365
- Goals, 155, 164, 291, 299
- Goals/models/tools, 10, 155
- Good figure, 319
- Good use of interactions, 162
- Goto-less, 276
- Granularity, 191
- Graph exchange language (GXL), 172, 366
- Graph rewriting, *see* Graph transformation
- Graph theoretical algorithm, 280
- Graph transformation, 265–266, 366
- Graphical user interface (GUI), 194
- Grep, 228
- Groomative, 30–31
- Groups, 29
- Guru, 243
- Habitability, 320
- Halpern, Mark I., 1, 44
- Halstead complexity (HC), 163
- Halstead, Maurice, 164
- Hardware maintenance, 2

- Hazard, 28, 366
- Hex-Rays Decompiler, 172
- Hierarchical algorithms, 281
- High cohesion, 16, 326
- High information content, 160–161
- High level of visibility, 70
- Higher order reuse cost model, 346, 348
- Historical co-change (HIS), 245
- Horizontal reuse, 366
- Horizontal traceability, 224, 229–230, 366
- Horseshoe, 366
- Horseshoe model, 138
- human resource, 43
- Hypertext markup language (HTML), 194
- Hypotheses, 14, 293–295, 299, 305, 308, 366
- Hypotheses strategies, 14
- I/O-emulator, 194
- Iceberg, 1, 92
- IDA Pro Disassembler and Debugger, 172
- Identification, 113
- IEEE 1042, 9
- IEEE/EIA 1219, 8, 94, 99
- Imagix 4D, 172
- Immediacy, 317
- Impact analysis (IA), 12, 93, 223, 225, 366
- Implement
  - change, 94
  - parallel operations and training, 110
  - state, 123
- Implementation, 97
  - details, 293
  - level, 136
  - model, 337
- Improvement algorithm, 281
- Improving maintainability, 134
- Improving quality, 134
- Incompatibility of upgrades, 63
- Increasing complexity, 3, 50–51, 134, 242
- Incremental
  - approach, 145, 233
  - development, 86, 366
- Indirect impact, 231
- Individuality, 160
- Information system (IS), 187, 366
- Information system gateway, 206–207, 366
- Information-flow metric, 164
- Initial development, 7, 87–88
- Initial entity, 244
- Initial reuse, 338, 339
- Input, 95
- Inspection, 276
- Integrated, 338
- Integrated metamodel, 312, 314
- Integration, 166
- Intention-based classification, 26
- Intentions, 317
- Inter-class coupling, 269
- Interface converter, 193
- Interface description languages (IDLs), 330
- Intermodule change propagation, 240
- Internal
  - documentation, 306
  - interface, 193
  - reuse, 366
  - semantic, 302
  - software value, 256
  - traceability, *see* Vertical traceability
- Internet Protocol (IP), 39
- Interpretation, 149
- Interprocedural analysis, 158
- Intra-class coupling, 269
- Intramodule change propagation, 238
- Intraprocedural analysis, 158
- Invariants, 265
- Inverse square law, 54–56
- Investigation activity, 40, 366
- Investigation report, 42
- IRAP (Input–Output Reengineering and Program Crafting), 172
- is-a, 294, 315
- ISO/IEC 12207, 8, 107
- ISO/IEC 14764, 8, 25–26, 99
- Isolation, 115, 148
- Iterative, 12
  - algorithm, 281
  - approach, 145
  - development, 86, 366
  - enhancement model, 85–86
  - method, 212
  - models, 7
  - process, 99
- JAD (JAVa Decompiler), 173
- Java 2 Enterprise Edition (J2EE), 201
- Jazayeri, Mehdi, 2

- Kaizen, 86
- kind-of, 294, 315
- Kitchenham, A, 28
- Knowledge, 291, 301–302
  - acquisition, 166, 291
  - base, 15, 298, 314, 366
  - elements, 294, 367
  - structure, 297, 309, 315, 367
- Koala, 332
- KobrA, 332
- Lack of cohesion in methods (LCOM), 163
- Law of
  - connectedness, 320
  - continuation, 320
  - contour, 320
  - evolution, 3
  - familiarity, 320
  - good figure, 320
  - orientation, 320
  - proximity, 320
  - similarity, 320
  - software evolution, 49
  - symmetry, 320
- Legacy, 187
- Legacy
  - DB, 214
  - DB manager, 214
  - information systems (LIS), 140, 188, 367
  - SampleData, 208
  - software, 11
  - state, 213
  - system, 11, 89, 187–188, 192, 195, 203, 367
- Legal, 328
- Lehman, Meir M., 1, 3, 44
- Lehman's law, 3, 51, 212
- Letovsky model, 298
- Leveraged, 338
- Lexical analysis, 10, 157, 367
- Library, 118
- License agreements, 165
- Lines of code (LOC), 163
- Linus's law, 59
- Linux, 4, 59–60
- LIS migration, 195–196
- Local area network (LAN) simulator, 260
- Locality, 321
- Logical
  - phase, 168
  - schema, 167–168, 195, 367
- Long-term memory a.k.a. knowledge base, 315
- Low coupling, 16, 326
- Low visual complexity, 160–161
- Macrostructure, 311
- Magnification function, 319
- Maintainability, 256, 263, 327
- Maintained
  - product, 37–38
  - product dimension, 37
- Maintainer, 2, 291, 367
- Maintenance, 1, 4, 25, 45
  - activities, 26
  - activities-based classification, 4
  - adaptive, 4
  - corrective, 4
  - cost, 327
  - of COTS-based systems, 5–6
  - engineer, 2
  - event, 42, 367
  - evidence-based classification, 5
  - factors, 37
  - iceberg, 1, 25
  - intention-based classification, 4
  - management, 42, 367
  - model, 84, 367
  - organization, 41–43
  - organization structure, 42, 367
  - perfective, 4
  - plan, 99, 101–102
  - preventive, 4
  - process, 37–38, 95, 100, 367
  - request, *see* Change request (CR)
  - review, 100, 105
  - staff, 44
- Make, 112, 115
- Management activity, 40, 367
- Managerial, 328
- Managers and policies, 274
- ManSART, 173
- Marr's theory, 319
- McCabe IQ, 173
- McIlroy, Dough, 15
- Mean time to maintenance request (MTMR), 215



- Mediator, 68
- Medium access control (MAC), 291
- Mental model, 14, 291, 293, 367
- Mental representation, 300
- Merging, 114
- Message
  - chain, 259
  - handler, 193
  - oriented middleware (MOM), 201
- Metadata, 214–215, 217
- Method, 41, 367
- Metric, 95, 265, 267, 368
- Metrication, 146
- Microstructure, 311
- Middleware, 64–65
- Migrate, 12, 188
- Migrate data, 216
- Migrating to a new technology, 134
- Migration, 12, 100, 106, 140–141, 368
  - of legacy information system, 140, 368
  - of LIS, 195
  - plan, 107
  - plan task steps, 108
  - planning, 196
  - process, 215
  - projects, 12
  - standard, 107
  - build, 70
- Minimal component coupling, 70
- Mishap, 28, 368
- Mission-critical
  - legacy system, 201
  - project, 196
- Model, 156, 368
- Model-driven engineering, 140, 368
- Modification activity, 40, 368
- Modification
  - implementation, 100, 105
  - of specification, 149
  - request (MR), 95, 102, 119, 368
- Modification request
  - analysis, 103
  - task steps, 104
- Modularity, 256
- Module
  - connection analysis, 238
  - level, 192
  - wrapper, 194
- Monolithic systems, 4
- Multi-agent, 53
- Multi-level, 53
- Multi-loop, 53
- Neighbors, Jim, 15, 325
- New DB, 214
- New DB manager, 214
- New state, 213
- Nondecomposable, 205–206
- Nondecomposable information system, 204, 368
- Non-functional requirements, 256, 368
- Normalization, 141, 148, 368
- Notification
  - of completion, 108, 110
  - intent, 107
  - of intent, 110
- Number of children (NOC), 163
- Obfuscation, 369
- Object
  - identification, 149
  - interpretation, 149
- Objectives, 134
- Observable behavior, 13
- Occurred set, 244
- Off-the-shelf component, 329
- Onion model, 58–59
- Open source community (OSC), 58
- Open source software (OSS), 368
- Openness of components, 67
- Operations, 312
- Opportunistic, 301, 337
- Optimization, 141, 369
- Optimization algorithm, 281
- Option task steps, 104
- Options, 104
- Organized reuse, 338–339
- Orientation, 320
- Original equipment manufacturer (OEM), 67
- OS/360 operating system, 54
- Output, 95
- Outsource, 11, 188
- Overloading, 195
- Paradigm, 41, 369
- Parallel operation, 196, 108
- Parameter coupling, 263–264, 269–270

- Parameterize methods, 273
- Paranas, David, 15, 325
- Parse tree, 157
- Parsing, 148
- Partial approach, 145
- Partially persistence, 166
- Path-based dynamic impact analysis, 235
- PBS (Portable Bookshelf), 173
- Pennington model, 310
- Peopleware, 43
- Perceptual
  - organization, 319
  - segregation, 319
  - theory, 319
- Perfective, 1, 289, 327
- Perfective
  - maintenance, 27, 49, 369
  - modification, 45
- Performance, 30, 31, 263
- Permanent files, 166
- Persistent data structure, 166
- Phase model, 150
- Phased interoperability, 196
- Phaseout, 8, 88–89, 187
- Physical phase, 168
- Physical schema, 168, 195, 369
- Plan, 171, 369
- Plan
  - change, 93
  - recognition, 171
- Planning, 117, 317
- Plans, 164, 293–294, 299, 315
- PLUS, 332
- Pollution, 53
- Poor lexicon, 54
- Portability, 15, 326
- Portfolio analysis, 12, 197, 369
- Postconditions, 265
- Post-delivery activities, 26
- Post-operation review, 109
- Practices, 276
- Precision, 226–227, 243–244, 246–247, 369
- Preconditions, 265
- Pre-delivery activities, 26
- Predicted set, 244
- Presence, 317
- Pretty printing, 257
- Preventive, 30–31
- Preventive maintenance, 27, 51, 137, 369
- Primary data, 212
- Primitive refactoring, 14
- Principle of
  - abstraction, 135, 369
  - alteration, 137, 142, 369
  - refinement, 135, 370
- Priority, 122
- Proactive, 327
- Probabilistic measures, 280
- Problem
  - domain, 299
  - domain knowledge, 311, 314
  - identification, 95
- Procedural
  - granularity, 148
  - level, 192
- Procedure, 41, 370
- Procedure wrapper, 194
- Process, 59, 144, 370
- Process
  - implementation, 100
  - improvement (how), 45
  - information, 246
  - level, 191
  - metrics, 230
  - model, 92, 144, 370
- Product, 37, 370
  - component, 39
  - development, 330
  - line development, 328–329
  - line, *see* Domain implementation
  - metrics, 230
  - quality, 40
  - reconfiguration, 68
  - upgrade, 37, 370
- Productivity, 327
- Program
  - analysis, 169, 370
  - analysis tool (PAT), 317
  - comprehension, 14, 93, 165, 291, 309–310, 370
  - conversion, 196
  - dependency, 224
  - dependency graph (PDG), 235, 370
  - documentation, 94
  - domain knowledge, 315
  - families, 15, 325, 370
  - fragments, 294
  - graph, 266

- knowledge, 317
- level, 192
- metrics, 162
- migration, 152, 165, 370
- model, 311, 313–314, 315
- model structure, 314
- plans, 171
- reading, 149
- rephrasing, *see* Rephrasing
- restructuring, *see* Restructuring
- reuse, 16, 326
- slices, 10
- slicing, 158, 282, 371
- translation, 141, 371
- understanding, *see* Program comprehension
- wrapper, 194
- Programmers, 274
- Programming
  - environment, 274
  - knowledge, 302, 308
  - plans, 294, 308, 315
  - rules of discourse, 308
- Project justification, 12
- Prolog, 262
- Properly nested programs, 277
- Proprietary software, 49
- Protocol
  - analysis, 315, 371
  - data, 315–316, 371
- Proximity, 319
- Pruning, 246
- P-type program, 47, 371
- PuLSE, 332
- Push down method, 267, 272
- Quality assessment, 167
- Quality assurance activity, 40, 371
- Query the tool first, 245
- Quick fix model, 85
- Random, 246
- Rationalization of control flow, 148
- Raymond, Eric S., 58–59
- Reachability
  - graphs, 232
  - matrix, 233
- Reactive, 327–328
- Read coupling, 163
- Readability, 256
- Real-world knowledge, *see* Domain knowledge
- RE-Analyzer, 173
- Recall, 226–227, 243–244, 246–247, 371
- Recency, 246
- Reclamation, 371
- Recode, 140, 143, 152, 371
- Reconstitute, 166
- Redesign, 141, 143, 371
- Redocumentation, 152–153, 372
- Redocumentation of artifacts, 154
- Reductive, 30–31
- Redundant structural data, 213
- Reengineered components, 214
- Reengineered state, 213
- Reengineering, 9–10, 133, 138–139, 154, 372
- Reengineering
  - assistant (RA), 174
  - process, 144
- Refactor, 258
- Refactoring, 13, 94, 141, 256, 259, 372
- Refinement, 135–136, 142
- Reflexive property, 240
- Reformative, 30–31
- Regeneration, 149
- Regression testing, 13, 94
- Rehosting, 140, 372
- Rejuvenation, 27, 372
- Release management, 57
- Releases, 60, 372
- Reliability, 16, 255, 257, 326–327
- Remote procedure calls (RPCs), 189
- Renovation, 141, 150–151
- Rephrasing, 140, 371
- Replace, 141, 143, 372
- Replace parameter with method, 272
- Replace strategy, 142
- Replacement strategies, 152
- Repository, 118, 372
- Repository database, 146
- Representation, 160
- Requirements level, 136
- Residual
  - data, 213
  - DB, 214
  - DB manager, 214
- Resilience to change, 161

- Resource, 372
  - constraints, 262
  - dependencies, 69
- Respecify, 141, 143, 372
- Response for a class (RFC), 163
- Restored components, 214
- Restored state, 213
- Restructuring, 13–14, 94, 137, 256–258, 372
- Restructuring algorithm, 142
- Rethink, 141, 143, 373
- Retirement, 100, 109, 376
- Retirement plan, 110
- Return value coupling, 263–264, 269
- Reusability, 16, 263, 373
- Reusable
  - artifacts, 326
  - assets, 15, 330
  - components, 347–348
  - software assets (RSA), 328–329
- Reuse, 289, 373
  - assets, 346
  - capability, 333–334, 373
  - capability model (RCM), 334–336
  - cost avoidance (RCA), 351
  - effectiveness, 333–334, 373
  - efficiency, 333, 373
  - library, 327, 329, 346, 349, 373
  - maturity model (RMM), 334, 373
  - opportunity, 333, 373
  - oriented model, 84
  - percent (RP), 351
  - proficiency, 333, 373
  - program, 329
  - value added (RVA), 351
- Reused source instruction (RSI), 350
- Reverse
  - engineering, 10, 135, 139, 153–154, 156, 164, 373
  - gateway, 204–206, 373
  - migration, 204, 373
- Review state, 122
- Review task steps, 106
- Revision control system (RCS), 112
- Rework, 141, 143, 373
- Rework strategy, 142
- Rewrite, 141, 143, 373
- Rewrite strategy, 142
- Rigi, 174
- Ripple effect, 13, 39, 93, 238, 374
  - analysis, 13, 225
  - for a program (REP), 242
  - and stability tool (REST), 242
- Ripple effects measurement, 242
- Ripple-sensitivity, 227, 374
- RiSE maturity model, 334, 338
- Risk, 327
- Risky, 57
- Robustness, 263
- Roll over, 212
- Royce, Winston, 6, 86
- RSEB, 332
- Rules of
  - discourse, 300, 315
  - programming discourse, 296, 374
- Safe, 57
- Safety constraints, 262
- Sample DataStore, 208–210
- Sandbox, 115
- Scalability, 263
- Scalability of visual complexity, 161
- Scattering and tangling, 374
- Schema(s), 293–294, 308, 315, 374
  - analysis, 166
  - conversion, 195
  - de-optimization, 170
  - integration, 169
  - translation and redesign, 166
  - untranslation, 170
- Scientific view point, 46
- Screen scrapping, 194, 374
- Script, 41, 374
- Scrubbing, 195
- Second law, 50–51
- SEELA, 174
- Self-regulation, 3, 50–51, 60, 242
- Semantic
  - distance, 319
  - knowledge, 301, 303
- Semantically redundant data, 213
- Semidecomposable, 204–205
- Semidecomposable information system, 204, 374
- Sequential dependency analysis, 261
- Service cost avoidance, 351
- Service-level agreement (SLA), 42–43, 374

- Service-oriented architecture (SOA), 42, 375
- Servicing, 8, 88–89
- Seventh law, 50, 53
- Severity, 122
- Shared variable coupling, 263–264, 269–270
- Sharpness, 227–228, 375
- Shipped source instruction (SSI), 350
- Shneiderman and Mayer Model, 301
- Short-term memory, 301, 303
- Side effect, 93, 154, 236, 375
- Similarity, 319
- Similarity metrics, 280
- Simple navigation, 161
- Single entity suggestions, 245
- Situation model, 310–311, 314–315
- Situation model structures, 314
- Sixth law, 50, 52
- Size, 38, 44, 263
- Slot-filler, 294
- Slot-type, 294
- Slot fillers, 315
- Slot types, 315
- Sneed, Harry M., 12
- Sockets, 189
- Soft-goal graph, 264, 375
- Software
  - agents, 53
  - aging, *see* Aging
  - architecture, 292
  - artifacts, 258
  - assessment, 167
  - asset, 326
  - certification laboratories (SCL), 64
  - clone, *see* Clone
  - configuration management (SCM), 9, 84, 111, 375
  - crises, 15
  - development, 26
  - development cost, 350
  - development error rate, 350
  - development life cycle (SDLC), 83–84
  - entropy, *see* Entropy
  - evolution, 375
  - geriatrics, 39
  - lifecycle objects (SLOs), 231–232, 376
  - maintainability, 26
  - maintenance, 2–3, 6, 25–26, 83–84, 376
  - maintenance life cycle (SMLC), 7
  - maintenance standards, 8
  - metric, *see* Metric
  - migration, 376
  - production consortium (SPC), 336
  - quality, 53
  - reclamation, 376
  - reconnaissance, 228
  - reengineering, 9–10
  - rejuvenation, 27
  - restructuring, 256, 275–276
  - retirement, *see* Retirement
  - reuse, 15, 325–326
  - specific knowledge, 291
  - structure, 273
  - value, 256
  - visualization, 135, 360, 317
- Soloway, Adelson, and Ehrlich model, 308
- Source code
  - control system (SCCS), 112
  - reengineering reference model (SCORE/RM), 146
- Source instruction reused by others (SIRBO), 350
- Source-to-source translation, 152
- SPE taxonomy, 46
- Specification
  - of actions, 149
  - of constraints, 149
- Speech sentences, 316
- S-Ratio, 228
- Stability
  - analysis, 238
  - measure, 238, 376
- Stable storage, 292
- Staged model, 7, 87, 187
- Staged model
  - for CSS, 87–88
  - for FLOSS, 90
- Stakeholders, 117, 198–199, 376
- Starting impact set (SIS), 93, 225, 228–231, 376
- State, 312
- Statecharts, 266
- Static
  - call graph, 158
  - elements, 293
  - program slice, 236

- Static (*continued*)
  - program slicing, 224
  - slice, 237
- Strategy, 295–296, 376
- Structural data, 213
- S-type program, 46, 376
- Submit state, 120
- Substitute algorithm, 271
- Suitability for automation, 161–162
- Support levels, 43
- Swanson, E. Burton, 1, 4
- Swanson's classification, 4
- Symmetric suggestions, 244
- Symmetry, 320
- Syntactic
  - analysis, 10, 157, 376
  - dependencies, 69
  - distance, 319
  - knowledge, 301–303
- System
  - monitoring, 66, 69
  - sandwich approach, 278
  - service wrappers, 191
  - test, 98
  - transition, 153
- Systematic reuse, 338, 340, 376
- Tailorability of components, 68
- Tailoring, 62, 376
- Target SampleData, 208
- Target system, 133–134, 142, 144, 208
- Target system testing, 150, 152
- Team structure, 58
- Technical, 328
- Technical dimension, 113
- Technique, 41, 276, 376
- Temporal constraints, 262
- TempStore (TS), 208–210, 212
- Termination-condition, 208–209, 212
- Test generation, 148–149
- Test Maturity Model (TMM), 64
- Test Process Improvement (TPI) model, 64
- Test verdict, 124
- Testability, 263
- Testing, 196, 262
- Testing and debugging, 66
- Textbase model, 310–311
- Text-structure, 293, 376
- Textual information, 246
- Theory of notation, 319
- Think aloud protocol (TAP), 316
- Thinking-aloud, 316, 376
- Third law, 50–51
- Threshold value, 208–209, 212
- Tools, 156, 274, 276
- Top-down, 301
- Top-down model, 313, 315
- Top-down structure, 314
- Torvals, Linus Benedict, 59
- Traceability, 224, 229
- Traceability analysis, 12
- Training, 30–31
- Training task steps, 108
- Transaction level, 192
- Transaction wrapper, 194
- Transformation, 148
- Transition net, 317
- Transitive, 240
- Transitive closure, 234
- Transmission control protocol (TCP), 292
- Trojan horse, 64
- Trouble shooting, 68
- Trunk, 114
- Tuval Software Industries, 174
- Two-phase commit protocol, 205
- Type decision question, 33
- Understandability, 16, 326
- Unix operating system, 115
- Unreliable COTS components, 64
- Unsafe, 57
- Updative, 30–31
- Usability, 377
- User, 377
  - interaction, 319
  - interface, 214, 216
  - manual, 94
  - request, 45
- Validation and verification, 149
- Variability, 44
- Variable name, 297
- Varying levels of detail, 161
- Verification, 103, 262
- Verification of hypothesis, 14
- Verification state, 124
- Version, 377
- Version control (VC), 113, 377

- Versioned staged model, 89–90
- Vertical reuse, 377
- Vertical traceability, 224, 230, 377
- Virtual Reality Modeling Language (VRML), 268
- Visualization, 160
- Walkthrough, 276
- Waterfall model, 6–7, 86
- Weighted methods per class (WMC), 162
- Weighted sum, 271
- Weiser, Mark, 158
- White-box reuse, 377
- White-box testing, 377
- Whole path profiling, 235
- Wide Spectrum Language (WSL), 172
- Wireless local area networks (WLAN), 292
- Workspace control, 115
- Wrap, 11, 188
- Wrapper, 11, 64–65, 189, 192
- Wrapping, 62, 195, 377
- Wrapping legacy systems, 12
- Write coupling, 163
- YACC, 157
- Zero–one (0–1) matrix, 239
- Zipf distribution, 246