

Index

- Abstraction 19, 20, 320, 510, 543
- Action plan, development of
 - corrective action 438
- Adaptation 58
- Adaptive maintenance 224
- Agents 532
- Agile methods 124
- Algorithmic cost estimation 464–465
- Algorithms 44
- Alternate releases strategy 552
- AMEISE 86–88, 91
- Analytical model 277–278
- Anomalous behaviour, causes of 131
- Anti-regressive activity 224
- Anti-regressive work 546
- Application
 - evolution of 27–28
- Architectural impact 288
- Architecture 77, 79–80, 88, 104–105, 269–280
 - and risk resolution 475
 - deterioration 227
- Architecture, system 105–106
- Artefact, technical 72
- As needed strategy 238
- Assembler 511
- Assignment 255
- Assumption 103, 105, 112, 113, 347–349
 - embedded in software artefacts 558
- Assumption reviews 554
- Average effort per release 133
- Balanced scorecards 487
- Baseline process 436
- Baselines 547
- Bayesian decision theory 520
- Behavioural stability 283
- Benchmarking 283, 306
- Benefits realization approach 472, 479
- Benefits-realized monitoring and control 479–481
- Bernoulli trial 149, 153
- BGW framework 285
- Binding 510
 - dynamic 512
 - dynamic vs static 510
 - static 510
- Biological evolution 194–197
- Black box modelling 465–466
 - control theory 461
- Black-box framework 285, 300
- Black-box models 559
- Blackmore 144
- Bounding 20–21
- Bricolage 191, 193
- Budgeted cost of work performed (BCWP) 477
- Budgeted cost of work scheduled (BCWS) 477
- Business case analysis 482–485
- Business event 405
- Business processes 54, 56, 60
- C++ 236, 514
- Calibration 277

- Call depths 236
- Calling structure 230
- Capability Maturity Model (CMM) 343, 428
- Capability Maturity Model Integration (CMMi) 343
- Capability/schedule/cost tradeoff analysis 471
- Cash receipts transaction management system 145, 148, 149
- Categorical data analysis 416–418, 420
- Causal models 493
- CeBASE 487
- CES. *See* Congruence Evaluation System
- Change 249–251, 254, 260, 265
 - gradual 75, 85
 - radical 78, 79, 85, 91
- Change drivers 75, 85
- Change impact analysis 283, 306
- Change logs 228
- Change rate 287, 288
- Change request 415, 417
- Change request forms 228
- Change request hierarchy 415
- Change ripple effects 238
- Change-prone modules 282, 305, 308
 - re-engineering of 381
- Chidamber & Kemerer 145
- Class of knowledge 407
- Class tree 149, 151–153
- Clean-up releases 548
- Closed source software 181
- Closed source systems 194, 200
- Closed-loop system 463
- Co-evolution 54, 56–58, 60, 199, 270
- COCOMO 464–465
- Code characteristics 414
- Code decay 226
- Code structure 212
 - code components 212
 - evolution of 213–218
 - folders and levels 212
- Commercial Off The Shelf (COTS) 12, 344, 346
- Communication gap 61
- Compiler 512–514
 - development, choice of function 513
- Complexity 53–68, 98, 99, 107, 109, 210, 219, 249, 261–263, 265, 545–547
 - measurement of 254, 257, 263
- Compliance 270–271
- Components 78, 79, 89
- Composition 318, 322–324, 334
- Computational models 199, 201
- Concept of large in software 8–10
- Confidence level 501
- Configuration 273–277
 - trade-offs in 275
- Conflicting evidence 495
- Conflicting requirements ‘issues’ 422
- Congruence 164
- Congruence Evaluation System (CES) 162–176
 - capabilities, analysis of 165
- Connectedness 57, 58, 67
- Connectors 84
- Conservation of familiarity 226, 263–264
- Conservation of organisational stability 263
- Constant effort law 80
- Constraint 400
 - mandated 407
- Constructive Cost Model II (COCOMO II) 444, 472
- Consumers 401–402
- Continuing change 261–262
- Continuing growth 264
- Control theory 461
- Control-flow 251, 260
- COPROMO 475
- Core Capability Demonstration (CCD) 482
- Corrective action 438
- Corrective maintenance 223
- Correctness 11, 13, 17–19, 29
- Cost 482–485
- Cost effective process 225
- Cost estimation 444, 493

- Cost factors 461
 - feedback-related 466–468
- COTS. *See* Commercial Off The Shelf
- Coupling 83–85, 90
- Cultural context 50

- Darwinian evolution 143, 144
- Data complexity 238, 244
- Data level 228, 236–240
- Data refinement 316
- Data-flow 251
- De-localisation 238, 240
- Declining quality 264
- Deliberate training 496
- Dependency check 422
- Derivation 319
 - feasibility 320
- Design 106, 121, 123
 - vs production 372
- Design history 314
- Design patterns 124
- Design processes 372
- Design recovery 219
- Design unit 78–79, 84, 88–90
- Detailed product tracing relationship 407
- Developers 400
- Development effort
 - distribution of 300
- Development process 411
- Developmental biology 194
- Diagnosis 520
- Dimensions of evolution 41–49
- Discrepancy reports (DRs) 415
- Discrete event simulation 431–432
- Discriminants 422
- Distribution model 153–159
- Document 399
- Documentation 227, 236
- Domains
 - nonparadigmatic 112
 - paradigmatic 113–114
 - pre-paradigmatic 113–116
 - real-world 105, 113
- Dominant design 195
- E*-type applications and software 12
- E*-type software 339, 344, 349, 350, 549
 - E*-type specification 350
- E*-type systems 75, 115–123, 161, 164, 171, 176, 278, 279, 446, 460
 - uncertainty 117–118
- Earned Value (EV) 471, 476–477
- Earned Value feedback process 476, 477
- Earned Value system 477
- Ecologies
 - informational 196
 - organizational 195, 199
- Ecosystem 57
- Effort estimation 497
- Elements ‘handled’ or ‘touched’ 213, 549
- Emergence 65, 66, 68
- Emergent properties 65, 67
- Empirical studies 30, 182–184, 207, 208, 210, 220
 - of software change processes 246
 - of software evolution 225
- Enabling environment 54, 64–68
- Engineering experiments 46
- Environment 161, 163
- Environmental changes 163, 168, 170, 171
- Environmental characteristics 170, 171
- Environmental evolution
 - impact of 167
 - requirements changes 171
- Ephemeral process 31
- Epistatic interaction 57, 67
- Error correction
 - cost of 423
- Estimation 459–468
- Estimation experience 490
- Estimation feedback 491, 493–497
- Estimation game 494
- Estimation learning 490–493
 - effort estimation 491
 - evaluation of outcome feedback 492
 - guidelines 493–497
 - motivation for 491

- Estimation learning (*continued*)
 timely feedback 492
 training opportunities 492
- Estimation models 444
- Estimation skills 494–495
- Evaluation 494
 of performance 494
- Evolution 270–271
 aspects of 71–91
 colloquial 75
 definition of 75
 generic theories of 96, 125
 in vitro 83
 in vivo 83
 inevitability of 13
 laws of 76, 82
- Evolution control 90
- Evolution drivers 76
- Evolution dynamics 9, 210
- Evolution identification method 287,
 305–307, 310
- Evolution management. *See* Evolution control
- Evolution model 194–197
- Evolution pattern 146–153
- Evolution planning 413
- Evolution studies
 open source 186–194
- Evolution theories 194–197
- Evolutionary biology 194
- Evolutionary burst 79, 84
- Evolutionary development 269, 271,
 273, 278, 548
- Evolutionary pattern 184–194
- Evolutionary processes 17
- Evolutionary spikes 80
- Evolutionary trends 215, 217
- Evolutionistic models 194
- Experience 500
- Experience factory 487
- Experience, forms of 44–47
 experimentation 46
 feedback 45
 understanding 46
- Expert estimation 489–503
- Expert judgment 491
- Exploration of the space of possibilities
 66, 68, 69
- Exponential 187
- F/OSS 181, 185–186
- Face recognition 528
- Failure occurrence 422
- Far-from-equilibrium 66–68
- Fault-prone modules 413
- FEAST 23, 367, 446
- FEAST hypothesis 556–557
- Feature interaction 329, 332
- Feature interaction, telephony 330
- Feature level 80
- Feature-oriented systems 329
- Feedback 45–46, 54, 56, 68, 107, 118,
 210, 219, 269, 275, 278, 279, 397,
 460–468, 507–535
 nonlinear 54, 56
 positive 103
 in simulation models 428
 sociology of 366
- Feedback characteristics
 taxonomy of 461
- Feedback control 277, 365–366, 373,
 472–476
 adequacy, assessment of 467
 mechanisms of 371
- Feedback control system 197–198
- Feedback cycle 472–476
- Feedback effects
 in software evolution 140
- Feedback loop 269, 277, 279
- Feedback overload 371
- Feedback properties 76
- Feedback system 264
- Feedforward capability 446
 role in risk management 453
- Feedforward model
 operational concept of 449
- Feedforward systems 447
- Fitness 161, 162
 external factors 161
 internal factors 161
 threats to 161
- Flexible architecture 269

- Folder structure evolution 207, 213–218
Formal estimation models 490
Formal methods 354
Formal specification 11, 557
Framework 85
Framework assessment 283, 309
Framework deployment 282, 306
Framework development effort 301, 304
Framework evolution 282, 284, 287, 291
Framework instantiation 282, 285, 300, 304, 307
Framework instantiation effort 304
Framework stability indicator 284, 298, 306, 307, 309
Free software 194
Function 512
Function call 510
Function level 228, 233–236
Functional requirements 407
Full Operational Capability (FOC) 482
- Gene 144
Global software process 106–108, 118–119
GNOME 186–188
GNU software 229
GNU/Linux 181, 186
Godfrey 186–188, 190
Goodness of fit 269
Granularity 228, 233, 236
GRAVA 522
 role of reflection in 527
Growing complexity 262–263
Growth rate 287, 288
Guarded commands 315
- Heat exchange simulation system 145, 148, 151, 152, 155
Hermeneutics 108–113
 methodological 109
 phenomenological 109, 114, 116
High-risk requirements 413, 414
Hindsight bias 495
- Impact analysis 250–251
Incremental effort 132
Influence
 vs control 370
Influencers 404
Information 74–75
Information content 74, 77, 83
Information technology 53–68
Infrastructure 195
Initial Operational Capability (IOC) 482, 483, 485
Instantiation effort 304, 307
Instantiation process 301
Intellectual reach 77
Intellectual span 77, 78, 81, 83
Intelligent room 531
Inter-release intervals 136
Interactions 272, 275, 277
Intermodule change propagation 256, 257
Interpretation 108–110
Interpretative processes 108, 109
Intramodule change propagation 255, 257
Invariant 325
Inverse square law 133, 136
IT legacy systems 53
 complexity of 53–68
- Judgmental bias 497
Jun 157
- Knowledge model 408
KnowledgePLAN 445
- Labeling program 523
Large software systems 8
Laws of software evolution. *See*
 Evolution, laws of
Laws of system evolution 132
Layered architecture 272
Learning biases 495
 over-confidence in own estimates 495
 reducing the impact
 conflicting evidence 495
Learning from experience 491

- Learning situations 496–497
- Legacy properties 225, 232, 242–244
- Legacy systems 53–68
- Legacy tendencies 225
- Lehman’s Laws 100–101, 116
 - of software evolution 346
- Libre 201
- Life Cycle Architecture (LCA) 482
- Life cycle objectives (LCO) 482
- Lines of code (LOC) 149, 211, 550
 - total number of 132
- Linker 511
- Linux Kernel 186, 188
- Lisp 512
- Load bearing walls 77, 79, 80
- Localised redevelopment 236
- LST paradigm 17–18

- Madey 190, 192
- Magic 319
- Maintenance 252–254
- Maintenance costs 53, 55
- Management planning and controlling 427, 432
- Management process 413
- Mature requirement 166, 177
- McCabe’s cyclomatic complexity 257
- Measurement 249, 261, 263
 - of ripple effect 250, 261, 262, 266
- Mechanisms 270–272, 279
- Mega-system 81, 85
- Memes 144, 194, 199
- Metagluе 532
- Methodological hermeneutics 110
- Metrics 146, 233, 242, 243, 250, 252, 253, 263
- Metrics data 416
- Metrics repository 429–431
 - transformation model 429
- Metrics-based assessment 283
- Minimum description length 524
- Model parameters
 - evolution of 155
- Model validation 131
- Model-Based (System) Architecting and Software Engineering (MBASE) 482

- Models 194–197
- Modification 322–324
- Module-clustering principles 138
- Modules 77
 - changes in 83
 - easy to modify 137, 138
 - evolution of 90
 - hard to modify 137, 138
 - relative frequencies of 137
 - total number of 132
- Monitoring 271, 274, 279
- Most likely effort 501
- Motivation 491, 494–495
 - for learning estimation skills 494–495
 - most likely effort 494
- Mozilla 181, 186, 191, 202

- Natural selection 144
- Negative binomial distribution 149, 153–155
- Networking 64, 65
- Nondeterminism 314
- Nonfunctional requirements 407
- Normal science 110–113
- Normalized-extent-of-change-metric 295, 297, 299

- Object-orientation 83, 90
- Object-oriented framework 281–309
- Observational level theory 344
- On-the-job feedback 491
- Ontology 197–200
- Open source software 181–200, 207, 540
- Open source systems 190, 196
- Open-loop 507
- Operation refinement 316
- Operational increments 416
- Order processing system 481–487
- Organisational context 50
- Organisational performance 468
- Organisational restructuring 59, 63
- Organisational structure 468
- OSS. *See* Open source software
- Outcome based control limits 433–434
- Over-confidence 499

- P*-type situations and software 13
- P*-type systems 120
- Paradigm 108, 110–115
 - characteristics of 111
 - formation processes 111
- Paradigmatic domain 113–114
- Parameterisation 85
- Parameters 263, 274–276
- Patchwork parser 525
- Pattern 75, 77, 80, 85
- Pearson's test 149
- Perfective maintenance 223
- Performance targets 435
- Persistence 272, 276
- Pervasive computing 530–535
 - intelligent room 531
- Policies 270, 271, 273, 276, 279
- Policy-checking mechanism 379–384, 389
- Postcondition 315
- Precondition 315
- Prediction interval 497
- Predictive power 134
- Predictive risk factors 422
- Preparation release 551
- Preventative maintenance 224
- Process evolution 28–32
- Process feedback 363–374, 428–431
- Process improvement 28–29, 342–343, 371, 556
- Process maturity 466, 475
- Process model evolution 32
- Process modelling techniques 459–468
- Process models 32, 272, 431
- Process regulation 366
- Process stability 424
- Process systems 367
- Process Tradeoff Analysis Method 427
- ProcessWeb 272–273, 277–278
- Producers 400–401
- Product lines 84, 195, 196
- Product-use case 405
- Productive unit 195
- Program classification 10
- Program conjunction 322
- Program disjunction 324
- Program modification 322–324
- Program refinement 314–320
- Program size distribution 153
- Program slicing 259
- Program volume 77
- Programming language 513
- Programming methodology 29
- Project drivers 399–405
 - influencers 404
- Project management 138, 140, 444
- Project planning 427
- Project sociology 405
- Proneness-to-change 283, 287
- Propagation 254–256
- Properties
 - structural. *See* Structural properties
- Prototyping 509
- Pseudo-time 136
- Quality gateway 408–409
 - testpoints 409
- Quality metrics 424
- Quantitative project management 431
- Rational Unified Process (RUP) 482
- Rationale 79
- Re-factoring 546
- Real earned-Value feedback control 478–481
- Real world 42–43
- Real-world program relationship 346–347
- Recovery 520
- Reengineering 229
- Reestimation 443–455
- Refactoring 150, 159, 191, 210, 224
- Refinement 17, 314–320, 325, 327
 - modification of 320
 - requirements 328
 - weakest preconditions 319
 - wide-spectrum language 314
- Reflections 71, 74, 76
- Regression coefficient 152, 153
- Relationship 407, 410
- Relative-extent-of-change-metric 293, 296, 297, 299

- Release 15, 16, 22
- Release planning 545
- Release Sequence Numbers (RSN) 132
- Reliability risk 411
- Replication 144
- Replicator 97–99, 116–117, 121, 123
- Representation 73, 326
- Requirements 103, 107, 109, 110, 113, 114, 116, 123, 397–410
 - environment-specific 166, 167
 - system-specific 166, 167
 - volatility of 82
- Requirements analysis 113–115
- Requirements attributes 414
- Requirements changes 161–177, 412
- Requirements component 407
- Requirements discovery
 - feedbacks in 397–410
- Requirements document 407
- Requirements evolution 162
- Requirements management 283, 307
- Requirements risk 415
- Requirements shell 409
- Requirements specification 397–399
 - feedbacks in 397–410
 - project constraints 397
 - testing requirements 397
- Requirements template 407
- Resource manager 315
- REST. *See* Ripple Effect and Stability Tool (REST)
- Restructuring 208, 210, 216, 217
- Results chain 479
- Retrenchment 324–335
 - within 331
- Returns on investment (ROI) 480
- Reusable asset 281, 309
- Reverse engineering 229
- Revolutionary modification 79
- Rigorous inspections 414
- Ripple 460
- Ripple effect 249–265
 - computation of 255
- Ripple Effect and Stability Tool (REST) 258
- Ripple effect measurement 250, 261, 262, 265, 266
- Ripple pattern 134, 136
- Risk analysis 420–423
- Risk factors 418–419
 - dependency check on 422
- Risk management 453–454
- Robustness 228
- Roles 272–273
 - scheduled 276
 - timed out 274, 276
- RUP. *See* Rational Unified Process
- S-curve 200
- S-type applications and software 10
- S-type systems 122
- Schedule As Independent Variable (SAIV) 477
- Scheduling 270–274
 - focus of 275
- Science
 - philosophy of 96, 108
- Scientific experiments 46
- Scientific knowledge 110, 121
- Scope of product 405
- Scope of work 405
- Securities management system 146
- Segmentation 523
- SEI Capability Maturity Models 475
- Self regulation 263, 547–549
 - establishing baselines 547
- Self-adaptive architecture 527
- Self-adaptive software 509, 517–521
 - agents 532
 - applications of 521–535
 - binding of function call to function value 510
 - internalized feedback 507–535
 - internalizing monitoring and evaluation functions 508
 - technology requirements
 - diagnosis 520
- Self-organisation 64, 65, 67, 68
- Semantic focus 72
- SESAM 86, 87, 91
- Seventh law, Lehman 171, 177

- Sierra mountainbikes 481
- Sierra's order processing systems 481
- Simula-67 513
- Simulated evolution 136
- Simulation 326
- Simulation modelling 460
- Simulation models 209, 431–432
- SLIM 445
- Smalltalk 147, 157, 513
- Social ecosystem 54, 57–58, 60
 - adaptation 58
- Software
 - categorisation of 77–82
 - definition of 72–75
 - as information 74–75
 - nature of 73, 74
 - open source 124
- Software adaptation 554
- Software aging 75
- Software application 543
- Software architecture 269, 270
- Software complexity 415
- Software component 99
- Software defects 552
- Software development
 - expert estimation of cost 489–503
 - feedback control of 472–476
 - process maturity 475
- Software development technologies 48
- Software development tools 48
- Software documentation 558
- Software ecology 515–517
- Software engineering 100, 116
 - methods 47–48
 - processes 47–49
 - techniques 48
- Software evolution 7–37, 96–104, 162, 171, 177, 208–210, 313–336, 368, 420, 460
 - dimensions of 41–49
 - domains 42–44
 - experience 44–47
 - process 47–49
 - empirical studies 30, 182–184
 - evidence of feedback
 - self-stabilization ripple 460
 - formal methods 313–336
 - global views of
 - process improvement 342–343
 - how of 341
 - laws of 100–101, 182–183, 260–265, 541–542
 - complexity of 545–547
 - self regulation 547–549
 - Lehman's laws of 30–31
 - management 539–560
 - monitoring of
 - policy-checking mechanism 379–384
 - retrenchment for 332
 - risk factors 226
 - stages in 16–17, 550
 - strata of 77
 - theories of 18–19, 99–101, 211, 339–355
 - good practice 345
 - scientific method 340
 - theory formation 345–346
 - what, why and how of 8, 539
 - why of 344
 - world view of
 - assumptions 347–349
- Software life cycle 245, 508
- Software maintenance 252–254
- Software maintenance models 252–254
- Software metrics 412, 549
- Software process 28, 35, 36, 373, 459–468
 - difficulties with feedback control 363–374
 - step functions vs regulation 371–372
 - feedback in 13–14, 23, 25, 33, 34
 - nature of 32
 - need for theory
 - formal methods 354
 - theory of 560
- Software process models 33
- Software quality 355, 553
- Software reestimation 443–455
- Software requirements analysis 418
- Software reuse 121–122, 124, 285

- Software size 412
- Software system 95–126
 - architecture of 104–105
 - assumptions about 113
 - environmental changes, impact of 162
 - evolution 270
 - and interpretation problem 108
 - interpretative processes 108
 - P* and *S* categories 120
 - real world 42
- Software system evolution 22–27
 - causal theory of 131
 - dynamic model of 131
- Software system growth 131–141
- Software systematics 198
- Software uncertainty
 - principle of 22, 346, 557–558
- Software, self-adaptive 507–535
- Software-related evolution
 - levels of 14–16
- Software-size 77
- Source code 249, 251
- Source code size 211
- Space shuttle flight software 418
 - complexity factors 419
 - criticality of change factors 419
 - locality of change factors 419
 - metric critical values 420
 - performance factors 420
 - personnel resources factors 420
 - requirements issues and functions factors 419
 - size factors 419
- SPE 95, 101–104
- SPE program classification 542–543
 - validation of 543
- SPE software classification 10
- SPE+ 96, 115–124
 - validation of 123
- Specification 101–104, 122–123, 350
- Specification statement 315
- Spherical model 138
- Splitting 233, 235, 236
- Sponsors 402–403
- Stability 254, 255, 265, 266
- Stability assessment 283, 284, 293–300, 306, 307
- Stability indicator 298
- Stakeholders 104–108, 399
 - policies of 123
 - satisfaction of 102, 117, 119, 120, 122
- Stakeholders' concerns 104, 119
- Standardisation 85
- Static binding 510–512
- Statistical clustering 526
- Statistical experiments 46
- Statistical modelling 143–149
- Statistical process control (SPC) 433
- Statistical significance 415
- Statistically invariant effort 133
- Step functions 371–372
- Stratification 72, 91
- Structural patterns 208, 213
- Structural properties 79
- Structural stability 283, 288
- Subject matter consultants 403
- Subroutine 511
- Subsystems 138, 140
- Success-critical stakeholders 479, 480
- Successive versions 228, 229
- Superlinear 186, 187, 193
- System 80
 - S*-type 101, 102–103, 122–123
 - E*-type 101, 103–104, 115–120
 - P*-type 101, 103, 120–122
- System architecture 105–106, 270, 271, 277
 - compliance of 270
 - ProcessWeb 272
- System complexity
 - growth of 133
- System dynamics 460, 548
- System function 270, 271
- System heterogeneity 138
- System identification 461–464
- System level 228, 230–233
- System size 132, 136, 139
- System volatility 412
- System-level evolution 84
- System-of-systems 80–81

- Systemic viewpoint 269, 270
- Systems-of-systems
 - evolution of 85
- Technical consultants 404
- Technical system 195
- Technological change 62
- Technological infrastructure 60–62
- Technology
 - vs sociology 366
- Teleological perspective 73
- Terminology
 - in software evolution 71
 - object-oriented 90
- Test 397
- Testable requirements 402, 405
- Testing 406
- Testpoints 409
- Theorem proving 526
- Theory evolution 43–44
- Theory formation 345–346
- Time of a release 132
- Timely feedback 492
- Total-extent-of-change 296
- Trade-offs 275–277
- Training opportunities 492
- Transformation 429

- Ubiquitous computing 530
- Uncertainty elicitation process 500
- Uncertainty, in *E*-type systems 117–118

- Undecidability 20
- Users 400, 401

- Validation 11, 545
- Validity
 - threats to 176
- Value 472, 476, 479
- Value-based feedback control 481–487
 - business case analysis
 - cost 482–485
- Value-based monitoring and control 485–487
- Variable definitions 254, 256–258
- Variable occurrence 256
- Verification 18–19
- Version 146, 150, 157
- Viewpoints, architectural 104, 106
- Virtual machine 272, 273, 275
- Vision system 522–528
 - image segmentation 523
- Visual builder 285
- Volatility management 84

- Waterfall model 508
- White-box framework 285
- White-box models 559
- Wicked system 75

- χ^2 test 149

- Z schema calculus 314