

Bibliography

- Abrial J-R (1996) *The B-book: assigning programs to meanings*. Cambridge University Press, New, NY
- Alderson A, Hull MEC, Jackson K, Griffiths LE (1998) Method engineering for industrial real-time and embedded systems. *Inf Softw Tech* 40:443–454
- Andriole SJ (1996) *Managing systems requirements: methods, tools and cases*. McGraw-Hill, New York
- Babich W (1986) *Software configuration management – coordination for team productivity*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA
- Bernstein P (1996) *Against the gods – the remarkable story of risk*. John Wiley & Sons Inc., Somerset, New Jersey, U.S.A
- Bjorner D, Jones CB (1978) *The Vienna Development Method: The Meta-Language, Lecture Notes in Computer Science*, Springer Verlag, Berlin, p 61
- Boehm B (1981) *Software engineering economics*, Prentice Hall PTR, Upper Saddle River, NJ, U.S.A
- Bolognesi T, Brinksma E (1987) Introduction to the ISO specification language LOTOS. *Computer Networks and ISDN Systems* 14(1):25–59
- Booch G (1994) *Object-oriented design with applications*. Benjamin Cummins, Redwood City
- Brown AW, Earl AN et al (1992) *Software engineering environments*. McGraw-Hill, London
- Budgen D (1994) *Software design*. Addison-Wesley.
- CarnegieMellon (2006) CMMI® for development, version 1.2. Software Engineering Institute, CMMI-DEV, Pitsburg., p 15213–3890
- Chaochen Z, Hoare CAR, Ravn AP (1991) A calculus of durations. *Inf Process Lett* 40(5):269–276
- Chen P (1976) The entity-relationship model—toward a unified view of data. In: *ACM transactions on database systems* 1/1/1976, ACM-Press. ISSN 0362-5915, S. 9–36.
- Clark KB, Fujimoto T (1991) *Product development performance*. Harvard Business School Press, Boston, MA
- Coad P, Yourdon E (1991a) *Object-oriented analysis*. Yourdon Press, Englewood
- Coad P, Yourdon E (1991b) *Object-oriented design*. Yourdon Press, Englewood
- Cooper RG (1993) *Winning at new products*. Addison-Wesley, Reading, MA
- Crosby PB (1979) *Quality is free*. McGraw-Hill, New York
- Crosby PB (1984) *Quality without tears*. New American Library, New York.
- Darke P, Shanks GG (1997) User viewpoint modelling: understanding and representing user viewpoints during requirements definition. *Inf Syst J* 7(3):213–240
- Davis AM (1993) *Software requirements: objects, functions and states*. Prentice-Hall, Englewood Cliffs

- DeGrace P (1993) *The Olduvai Imperative: CASE and the state of software engineering practice*. Yourdon Press, Englewood Cliffs, NY
- DeMarco T (1978) *Structured analysis and system specification*. Yourdon Press, Englewood Cliffs, NJ
- DeMarco T (1982) *Controlling software projects*. Yourdon Press, Englewood Cliffs, NJ
- DeMarco T, Lister T (1987) *Peopleware – Productive projects and teams*. Dorset House Publishing, New York
- Dick J, Faivre A (1993) Automating the generation and sequencing of test cases from model-based specifications. In: Conf. Proc. FME '93, Odense, Denmark, April 1993.
- Dick J, Jones B, (2012) On the Complexity of Requirements Flow-down Structures. INCOSE International Symposium 22 (1):1197–1206
- Dick J, Llorens J (2012) Using statement-level templates to improve the quality of requirements. In: Proc. 24th Int. Conf. on software & systems engineering and their applications, Paris, October 2012.
- Easterbrook S, Nuseibeh B (1996) *Using viewpoints for inconsistency management*. *Softw Eng J* 11(1):31–43
- Faulk S, Brackett J, Ward P, Kirby J (1992) *The core method for real-time requirements*. *IEEE Softw* 9(5):22–33
- Finkelstein A, Kramer J, Nuseibeh B, Goedicke M (1992) Viewpoints: a framework for integrating multiple perspectives in systems development. *Int J Softw Eng Knowl Eng* 2(10): 31–58
- Fowler M, Scott K (1997) *UML distilled: applying the standard object modeling language*. Addison-Wesley, Reading
- Gilb T (1988) *Principles of software engineering management*. Addison-Wesley, Reading, MA
- Gilb T (2005) *Competitive engineering: a handbook for systems engineering, requirements engineering and software engineering management using planguage*. Elsevier Butterworth-Heinemann, Oxford
- Gorchels L (1997) *The product manager's handbook*. NTC Business Books, Chicago, IL
- Gotel OCZ, Finkelstein ACW (1995) *Contribution structures*. Proceedings RE'95. IEEE Press, York, UK
- Harel D (1987) Statecharts: a visual formalism for complex systems. *Sci Comput Program* 8:231–274
- Hull MEC, Taylor PS, Hanna JRP, Millar RJ (2002) Software development processes – an assessment. *Inf Softw Tech* 44(1):1–12
- Humphrey WM (1989) *Managing the software process*. Addison-Wesley, NY
- IEEE STD 1220-1998 (1998) *Standard for application and management of the systems engineering process*. IEEE, New York
- INCOSE (2007) INCOSE SE Vision 2020, INCOSE-TP-2004-004-02. In: International council on systems engineering (INCOSE) 7670 opportunity road, Suite 220, San Diego, California 92111-2222 USA.
- INCOSE (2017) Guide for writing requirements, INCOSE-TP-2010-006-02, version 2.1, 2017, prepared by requirements working group. In: International council on systems engineering (INCOSE) 7670 opportunity road, Suite 220, San Diego, California 92111-2222 USA.
- Jackson M (1995) *Software requirements & specifications: a lexicon of practice, principles and prejudices*. Addison-Wesley, Wokingham, UK
- Jacobsen I, Christerson M et al (1993) *Object-oriented software engineering*. Addison-Wesley, Wokingham, UK
- Kennedy L (2005) *Keeping the promise: a work ethic for doing things right*. Reliable Man Books, Orlando, FL
- Kotonya G, Sommerville I (1996) Requirements engineering with viewpoints. *Softw Eng J* 11(1):5–11
- Kotonya G, Sommerville I (1998) *Requirements engineering: processes and techniques*. John Wiley & Sons, Chichester

- Leite JCP, Freeman PA (1991) Requirements validation through viewpoint resolution. *Trans Softw Eng* 12(2):1253–1269
- Loucopoulos P, Karakostas V (1995) *Systems requirements engineering*. McGraw-Hill, London
- Mazza C et al (1994) *ESA – Software engineering standards*. Prentice-Hall, New Jersey
- Monroe RT, Kompanek A, Metlon R, Garlan D (1997) Architectural styles, design patterns, and objects. *IEEE Software* 14(1):43–52. Jan.
- Mumford E (1989) User participation in a changing environment – Why we need it. In: Knight K Ed *Participation in systems development*. Kogan Page, London
- Nuseibeh B, Kramer J, Finkelstein A (1994) A framework for expressing the relationships between multiple views in requirements specification. *Trans Softw Eng* 20(10):760–773
- Oliver DW, Kelliher TP, Keegan JG (1997) *Engineering complex systems with models and objects*. McGraw-Hill, New York
- OMG (2003) *The unified modelling language version 2*, www.omg.org.
- Page-Jones M (1980) *The practical guide to structured systems*. Yourdon Press, NY
- Perrow C (1984) *Normal accidents*. Basic Books, NY
- Petroski H (1982) *To engineer is human – the role of failure in successful design*. Vintage Books, NY
- Petroski H (1996) *Invention by design: how engineers get from thought to thing*. Harvard University Press, Cambridge, MA
- Poots C, Takahashi K et al (1994) *Inquiry-based requirements analysis*. *IEEE Softw* 11(2):21–32
- Pressman RS (1997) *Software engineering: a practitioner's approach*. McGraw-Hill, London
- Ross DT (1977) Structured analysis (SA): a language for communicating ideas. *IEEE Trans Softw Eng* 3(1):16–34
- Ross DT (1985) Applications and extensions of SADT. *IEEE Comput* 18(4):25–34
- Ross DT, Schoman KE (1977) Structured analysis for requirements definition. *IEEE Trans Softw Eng* 3(1):6–15
- Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorenzen W (1991) *Object-oriented modeling and design*. Prentice-Hall, NY
- Shlaer S, Mellor SJ (1991) *Object life cycles – modeling the world in states*. Yourdon Press, NY
- Shlaer S, Mellor SJ (1998) *Object-oriented systems analysis*. Prentice-Hall, Englewood Cliffs
- Software Engineering Institute (Carnegie-Mellon) (1991) *Capability maturity model for software*. In: *Tech. report CMU/SEI-91-TR-24*.
- Sommerville I (1996) *Software engineering*. Addison-Wesley, Wokingham
- Sommerville I, Sawyer P (1997) *Requirements engineering: a good practice guide*. John Wiley and Sons, Chichester
- Spivey JM (1989) *The Z notation: a reference manual*. Prentice Hall International Series in Computer Science, Hemel Hempstead, Hertfordshire, UK
- Standish Group (1995), *The CHAOS Report 1995*, Standish Group International. Inc., Boston MA
- Standish Group (2015), *The CHAOS Report 2015*, Standish Group International. Inc., Boston MA
- Stevens R, Brook P, Jackson K, Arnold S (1998) *Systems engineering: coping with complexity*. Prentice-Hall, Europe
- SyntheSys (2014) SPIRIT – UML modelling, white paper, http://www.synthesys.co.uk/brochures_new/5.4SPIRIT%20Modelling%20Tech%20Sheet.pdf.
- Wheatcroft L, Ryan M, Dick J (2016) On the use of attribute to manage requirements. *Syst Eng J Pub. Wiley* 19(5):448–458
- Woodcock H, Chard J (2013) A proposed Agile systems engineering manifesto. In: *INCOSE UK ASEC 2013*. <https://www.slideshare.net/hazel-woodcock/a-proposed-agile-systems-engineering-manifesto-noteless>.
- Yourdon EN (1990) *Modern structured analysis*. Prentice-Hall, Englewood Cliffs, NY
- Zave P (1997) Classification of research efforts in requirements engineering. *ACM Comput Surv* 29(4):315–321

Index

A

- Abstract model, 64
- Abstraction, 22, 57, 96, 108
- Acceptance criteria, 38–40, 133
- Acquisition, 3, 4–9
- Agile development, 21–22
- Agnostic design, 23
- Allocation
 - functional, 20, 102, 172
 - non-functional, 20, 172
- Analysis
 - coverage, 17, 161
 - derivation, 16, 52–54, 139–144, 152–153, 156, 161, 167
 - impact, 16, 161, 181, 183
 - rich traceability, 170–171
- Assessment, 97, 101
- Assumptions. *See* domain knowledge
- Atomicity, 108, 111
- Attribute, 97, 97–98
 - categories, 98
 - rationale, 162, 165

B

- Background information, 96
- Bid management, 9–13
- Black box, 23
- B-Method, 89
- Boilerplates, 103, 103–105, 104, 105
- Brainstorming, 71, 111, 117
- Business objectives, 14

C

- Capability, 102
- Capacity, 102, 105

Change

- complexity, 184
- estimation, 184
- impact, 16, 161, 181, 183
- latent, 183, 184
- management, 1–2, 13, 14, 183
- request, 184

Clarity, 107

Class Diagram, 67, 82

Classification

- primary, 95, 100
- secondary, 95, 100

Communication, 12, 19

Completeness, 108

Complexity, 1

Compliance statement, 174

Confidential information, 103

Conflicts, 99–100

Consistency, 99–100, 108

Constraints, 102, 104, 144

Context diagram, 86

Contextual information, 94

COTS, 1

Coverage analysis, 17, 161

Cultural change, 22

Cup, 6

D

data dictionary, 59

Data flow, 43, 58–64, 59

Data flow diagram, 58, 96

data store, 59

data transformation, 59

Decomposition

- functional, 20, 102, 172

- Decomposition (cont.)
 - non-functional, 20, 172
 - not necessary, 172
- DeMarco, 69
- Derivation analysis, 161, 167
- Design agnosticism, 23
- Design freedom, 22, 101
- Design intent, 159, 186
- Design justifications. *See also* satisfaction
 - statement, 165, 174
- Development
 - change, 41–42
 - ideal, 40–41
 - process, 35, 59
 - systems, 33–36, 102
- Diagram
 - activity, 73
 - block definition, 86
 - class, 67
 - context, 86
 - data flow, 58
 - entity-relationship, 64
 - ERD, 64
 - parametric, 86
 - requirements, 86, 170
 - sequence, 85
 - state transition, 65
 - statecharts, 65
- Domain knowledge, 96, 167
- DOORS, 187
 - architecture, 188–189
 - export, 205
 - import, 205
- E**
- Elementary traceability, 161, 162
- Emergent properties, 5, 6, 19, 172
- Entity-relationship diagram, 64
- Environment, 7
- ERD, 64
- Estimation, 1
- Example, 167
 - aircraft engine, 108
 - banking, 144–147
 - barrista, 162, 163
 - car, 147–152
 - cooker, 161
 - cup, 6
 - electro-chlorination unit, 172
 - military vehicle, 161
 - New York underground, 167
 - nightmare, 108
 - RAC rally, 11
 - sailing boat, 121, 188
 - timetables, 168
 - vehicle, 164
- external entity, 59
- F**
- Feasibility, 107
- Feedback, 111
- Filtering, 103
- Formal method, 89
- Functional allocation, 172
- Functional decomposition, 172
- Functional requirements, 102
- Functionality
 - human interaction, 142, 146, 150
 - interface, 141–142, 145–146, 149–150
 - internal, 140–141, 144–145, 149
 - safeguard, 142–143, 146–147, 150–151
- G**
- Generic process, 49–56
 - instantiation, 114–115, 136, 154
- H**
- Hierarchies, 95, 96
- Human interaction, 142–143, 142, 146
- I**
- Impact analysis, 13, 16, 161, 181
- Impact assessment, 184
- INCOSE, 98
- Incremental development, 21
- Information model, 43–49
- Intentional design, 186
- Interface register, 29, 30
- Interfaces, 5, 6, 24, 29
 - decomposition, 29
- Invitation To Tender, 10
- J**
- Justification, 162
- K**
- Key requirements, 96–97
- L**
- Language, 101, 101–103
- Legality, 107
- Let-out clauses, 108
- Lifecycle, 11–14
- LOTOS, 89

M

- Management, 1–2
 - experience, 2, 3
 - of change, 4, 7–9, 12–13, 15, 20–21, 24
 - or bids, 9–13
 - planning, 4–7, 10–11, 13–14, 15–19, 23–24
 - problems, 2–4
- MBSE, 91
- Method, 68
- Metrics, 179–185, 186
 - global, 180
 - imbalance, 182, 182–183
 - phase, 180
- Model-based System Engineering, 91
- Modelling
 - and qualification, 91
 - class diagram, 82
 - data flow diagram, 58
 - parametric diagram, 86
 - requirement diagram, 86
 - sequence diagram, 85
 - SysML, 86
 - use case diagram, 83
- Models
 - abstract, 64
 - system, 23, 69, 101
- Modes of operation, 144, 147, 151–152
- Modularity, 108
- Monitoring progress, 4, 7, 11–12, 14–15, 19–20, 24, 179–185

N

- Necessity, 169
- Non-functional allocation, 172
- Non-functional decomposition, 172

O

- Object Management Group, 86
- Operability, 105
- Operational modes, 108
- Organisations
 - acquisition, 3, 4–9
 - product developers, 3, 15–21
 - suppliers, 3, 9–15
 - types, 3

P

- Paragraph numbers, 93
- Parametric diagram, 86
- Patterns
 - flow-down, 159, 165, 170, 172

PBS, 26

- Performance, 20, 102
- Performance attribute, 102
- Periodicity, 103, 105
- Praxis Critical Systems, 165
- Precision, 108
- Problem
 - versus solution, 3, 21–23
- Problem domain, 21, 22, 101, 113–114
- Process
 - analysis, 15, 51–52, 170–171
 - derivation, 16, 52–54, 139–144, 152–153, 156, 161, 167
 - generic, 36–40, 43–49, 49–56
 - modelling, 51–52
 - qualification, 54–56, 46–47, 108
 - requirements agreement, 49–51
- Process specification, 59
- Product breakdown structure, 26
- Product developers, 3, 15–21
- Product distribution, 1
- Product families, 12, 17–19, 18, 20
- Programme management, 12
- Progress measurement, 17
- Project
 - decision factors, 2
 - plan, 1
- Project failure, 2
- Project success, 3
- Propositional logic, 171
- Prototypes
 - requirements from, 130–131

Q

- Qualification, 20, 108
 - strategy, 132, 133–134
- Qualification argument. *See* qualification statement
- Qualification rationale. *See* qualification statement
- Qualification statement, 173
- Qualification strategy, 21, 38–40
- Quality, 11, 10–11
 - factors, 98

R

- Reality, 1
- Redundancy, 108
- Request For Proposals, 10
- Requirement
 - definition, 7–8
 - requirement diagram, 86

Requirements

- abstraction, 108
- agreement, 45–46, 49–51, 116, 153
- allocation, 152–153, 165
- atomicity, 108, 111
- attributes, 97, 97–98, 98
- boilerplates, 103, 103–105, 104, 105
- capabilities, 104
- capture, 126–132
- clarity, 107
- classification, 95
- completeness, 108
- conflicting, 95, 99–100
- consistency, 99–100, 108
- constraints, 105, 131–132
- criticality, 183
- decomposition, 181, 185
- derivation, 16, 52–54, 139–144, 152–153, 156, 161, 167
- derived, 38
- documents, 94–96
- elaboration. *See* satisfaction statement
- expressing, 103
- feasibility, 107
- flow-down, 159, 165, 170, 172
- functional, 20, 102, 172
- identification, 117–118, 128
- identifying, 93, 101
- identity, 165, 166
- imbalance, 182, 182–183
- input, 38
- language, 101, 101–103, 102, 103, 107–109, 170
- legality, 107
- management, 1–2, 13, 14, 183
- modularity, 108
- nightmare, 108
- non-functional, 20, 172
- performance, 102
- precision, 108
- qualification, 46–47, 54–56, 108
- redundancy, 108
- refinement, 132
- reuse, 95
- reviewing, 94
- satisfaction, 47–48, 108
- singular, 107
- stakeholder, 23, 101, 102
- stakeholder v. system, 23
- structure, 94–96, 108, 111, 152
- structuring, 95, 122–126
- system, 23, 33–36, 69, 101, 102
- trade-off, 101

- uniqueness, 107
- value of, 100, 100–101, 101
- verifiability, 108
- workshops, 129–130
- writing, 94

- Requirements engineering
 - definition, 5, 8, 9–10

- Requirements expression, 97, 103

- Requirements flow-down, 159

- Requirements statement, 97

- Reuse, 12, 103

- Review process

- criteria, 94

- requirements, 86, 170

- traceability, 170

- Rich traceability, 159, 165, 167, 197, 200

- analysis, 15, 51–52, 170–171

- implementation, 174

- multi-layer, 166, 169, 174

- single-layer, 174

- Risk management, 2

- Rumbaugh, 69

S

- Satisfaction, 108

- Satisfaction argument, 163, 165, 167, 170

- language, 101, 101–103, 102, 103, 107–109, 170

- Satisfaction intent, 173

- Satisfaction rationale, 163

- Satisfaction statement, 162–163, 163, 167, 185

- disjunction, 164

- Scenarios, 20, 96

- Sequence diagram, 85

- Shlaer Mellor, 69

- Singularity, 107

- Software, 1

- Solution

- versus problem, 3, 21–23

- Solution domain, 21, 22, 101, 102, 135–137

- Sorting, 103

- Specification

- process, 35, 59

- Speculation, 108

- Stage gate, 16

- Stakeholder, 102

- constraints, 105, 131–132

- definition, 5, 8, 9–10

- Stakeholder requirements, 23, 101, 102

- Stakeholders, 94

- identification, 117–118, 128

- interviewing, 126–127

- State charts, 96
 - State transition diagram, 65
 - Statechart, 20, 65
 - Strategy. *See* satisfaction statement
 - Structure, 108
 - Subsystems, 154–156
 - Sufficiency, 169
 - Suppliers, 3, 9–15
 - SysML, 86
 - System*
 - definition*, 5, 8, 9–10
 - scope*, 122
 - System architecture, 154–156
 - System model, 137–138
 - System modelling, 18–20, 19, 20, 69, 96, 167
 - System requirements, 23, 101
 - System transaction, 143, 147, 151
 - Systems development, 33–36
 - Systems engineering, 4–7
 - Systems of systems, 5
 - Systems requirements, 102
- T**
- Tender assessment. *See* assessment
 - Test planning, 17
 - Time to market, 2
 - Timeliness, 105
 - Traceability, 13, 14–17, 17
 - analysis, 15, 51–52, 170–171
 - benefits, 185, 186
 - breadth, 180, 180–181
 - depth, 180, 181
 - downwards, 183
 - effort, 185
 - elementary, 160–162, 161, 162
 - growth, 180, 181–182
 - imbalance, 182, 182–183
 - matrix, 160
 - metrics, 179–185
 - multi-layer, 166, 169, 174
 - overhead, 17
 - rationale. *See* satisfaction statement
 - rich, 159, 165, 167
 - upwards, 183
- U**
- UML, 81
 - class diagrams, 20
 - message sequence charts, 20
 - Unified Modelling Language, 81
 - Uniqueness, 107
 - Use case diagram, 83
 - Use scenarios
 - characteristics, 120–122
 - creation, 118–122
- V**
- Value function, 100, 101
 - Variants, 17–19, 18, 20
 - VDM, 89
 - Verifiability, 108
 - Versions, 17–19, 18, 20
 - VISION, 169
 - V-Model, 11
- W**
- Waterfall development, 22
 - West Coast Mainline, 4, 165
 - Word processor, 93
- Y**
- Yourdon, 69
- Z**
- Z, 89