

Part I: The Fundamentals

Part II: Requirements Engineering Practices

Part III: Enablers and Stumbling Blocks

Conclusions

References

Further reading

- I. Alexander, R. Stevens (2002). *Writing Better Requirements*. London: Addison-Wesley.
- A. Davis (2005). *Just Enough Requirements Management*. New York: Dorset House.
- D.C. Gause, G.M. Weinberg (1989). *Exploring Requirements: Quality before Design*. New York: Dorset House.
- M. Glinz (2016). *How much Requirements Engineering do we need?* Softwaretechnik-Trends **36**(3). 19–21.
- M. Glinz (2017). *A Glossary of Requirements Engineering Terminology*, Version 1.7. International Requirements Engineering Board (IREB). <https://www.ireb.org/en/downloads/#cpre-glossary>
- E. Gottesdiener (2002). *Requirements by Collaboration: Workshops for Defining Needs*. Boston: Addison-Wesley.
- M.A. Jackson (1995). *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. Addison-Wesley (ACM Press books): Wokingham, etc.
- A. van Lamsweerde (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Chichester: John Wiley & Sons.
- S. Lauesen (2002). *Software Requirements: Styles and Techniques*. London: Addison-Wesley.
- K. Pohl (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin: Springer.
- K. Pohl, Rupp, C. (2015). *Requirements Engineering Fundamentals*, 2nd edition. Rocky Nook. Distributed by O'Reilly Media, Sebastopol, CA.
- S. Robertson, J. Robertson (2012). *Mastering the Requirements Process*. 3rd edition. Boston: Addison-Wesley.
- K. Wiegers, J. Beatty (2013). *Software Requirements*. 3rd edition. Redmond, Wa.: Microsoft Press.

References

- J.-R. Abrial (2009). *Modelling in Event-B: System and Software Engineering*. Cambridge: Cambridge University Press.
- I. Alexander, R. Stevens (2002). *Writing Better Requirements*. London: Addison-Wesley.
- P. G. Armour (2004). *The Laws of Software Process: A New Model for the Production and Management of Software*. Boca Raton, Fl.: CRC Press.
- D. M. Berry (2002). Formal Methods: The Very Idea. Some Thoughts About Why They Work When They Work. *Science of Computer Programming* **42**(1):11–27.
- B. Boehm (1981). *Software Engineering Economics*. Englewood Cliffs: Prentice Hall.
- D. Björner, C. Jones (1978). *The Vienna Development Method*. Berlin: Springer.
- G. Booch (1994). *Object Oriented Analysis and Design with Applications*. Second Edition. Redwood City, Ca.: Benjamin/Cummings.
- G. Booch (1986). Object-Oriented Development. *IEEE Transactions on Software Engineering* **12**(2):211–221.
- T. Bruckhaus, N. Madhavji, I. Janssen, J. Henshaw (1996). The Impact of Tools on Software Productivity. *IEEE Software* **13**(5):29–38.
- J. Carroll (1995). The Scenario Perspective on System Development. In J. Carroll (ed.): *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York: John Wiley & Sons. 1–18.
- P. P. Chen (1976). The Entity-Relationship Model -Toward a Unified View of Data. *ACM Transactions on Database Systems* **1**(1):9–36.
- A. Cockburn (2001). *Writing Effective Use Cases*. Boston: Addison-Wesley.

References – 2

- M. Cohn (2004). *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley.
- T. DeMarco (1978). *Structured Analysis and System Specification*. New York: Yourdon Press.
- J. Dick (2005). Design Traceability. *IEEE Software* **22**(6):14–16.
- O. Dieste, N. Juristo, F. Shull (2008). Understanding the Customer: What Do We Know about Requirements Elicitation? *IEEE Software* **25**(2):11–13.
- M. E. Fagan (1986). Advances in Software Inspections. *IEEE Transactions on Software Engineering* **SE-12**(7):744–751.
- D. P. Freedman, G. M. Weinberg (1982). *Handbook of Walkthroughs, Inspections and Technical Reviews*. 3rd edition. Boston, Toronto: Little, Brown and Co.
- D. C. Gause, G. M. Weinberg (1989). *Exploring Requirements: Quality before Design*. New York: Dorset House.
- J. M. Carrillo de Gea, J. Nicolás, J. L. Fernandez Alemán, A. Toval, C. Ebert, A. Vizcaíno (2011). Requirements Engineering Tools. *IEEE Software* **28**(4):86–91.
- M. Glinz (1995). An Integrated Formal Model of Scenarios Based on Statecharts. In W. Schäfer and P. Botella (eds.): *Software Engineering – ESEC '95*. Proceedings of the 5th European Software Engineering Conference. Berlin, etc.: Springer. 254–271.
- M. Glinz (2000a). Improving the Quality of Requirements with Scenarios. *Second World Congress on Software Quality*. Yokohama. 55–60.

References – 3

- M. Glinz (2000b). Problems and Deficiencies of UML as a Requirements Specification Language. *Tenth International Workshop on Software Specification and Design*. San Diego. 11–22.
- M. Glinz, S. Berner, S. Joos (2002). Object-Oriented Modeling with ADORA. *Information Systems* **27**(6):425–444.
- M. Glinz (2005). *Software Engineering I* [in German]. Lecture notes, University of Zurich.
- M. Glinz (2005). Rethinking the Notion of Non-Functional Requirements. *Third World Congress for Software Quality (3WCSQ 2005)*, München, Vol. II. 55–64.
- M. Glinz (2007). On Non-Functional Requirements. *15th IEEE International Requirements Engineering Conference (RE'07)*, Delhi, India. 21–26.
- M. Glinz, R. Wieringa (2007). Stakeholders in Requirements Engineering. *IEEE Software* **24**(2):18–20.
- M. Glinz (2008). A Risk-Based, Value-Oriented Approach to Quality Requirements. *IEEE Software* **25**(2):34–41.
- M. Glinz, S. Fricker (2013). On Shared Understanding in Software Engineering. *GI Conference Software Engineering 2013, Lecture Notes in Informatics vol P-213*. 19–35.
- M. Glinz (2014). *A Glossary of Requirements Engineering Terminology*, Version 1.6. International Requirements Engineering Board (IREB). <https://www.ireb.org/de/downloads/#cpre-glossary>
- M. Glinz (2016). *How much Requirements Engineering do we need?* *Softwaretechnik-Trends* **36**(3):19–21.
- M. Glinz, S.A. Fricker (2015). On Shared Understanding in Software Engineering: An Essay. *Computer Science – Research and Development* **30**(3-4):363–376.

References – 4

- R. Goetz, C. Rupp (2003). Psychotherapy for System Requirements. *2nd IEEE International Conference on Cognitive Informatics (ICCI'03)*, London, UK. 75–80.
- J. Goguen, C. Linde (1993). Techniques for Requirements Elicitation. *First IEEE International Symposium on Requirements Engineering (RE'93)* San Diego, California, USA. 152–164.
- T. Gorschek, S. Fricker, K. Palm, S. A. Kunsman (2010). A Lightweight Innovation Process for Software Intensive Product Development. *IEEE Software* **27**(1):37–45.
- O. Gotel, A. Finkelstein (1994) An Analysis of the Requirements Traceability Problem. *First International Conference on Requirements Engineering*, Colorado Springs. 94–101.
- O. Gotel, J. Cleland-Huang, J. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, P. Mäder (2012). Traceability Fundamentals. In J. Cleland-Huang, O. Gotel, A. Zisman (eds.): *Software and Systems Traceability*. London: Springer. 3-22.
- E. Gottesdiener (2002). *Requirements by Collaboration: Workshops for Defining Needs*. Boston: Addison-Wesley.
- C. A. Gunter, E. L. Gunter, M. Jackson, P. Zave (2000). A Reference Model for Requirements and Specifications. *IEEE Software* **17**(3):37–43.
- P. Grünbacher, N. Seyff (2005). Requirements Negotiation. In A. Aurum, C. Wohlin(eds): *Engineering and Managing Software Requirements*. Springer. 143-162.
- D. Harel (1988). On Visual Formalisms. *Communications of the ACM* **31**(5):514–530.
- A. M. Hickey, A. M. Davis (2003). Elicitation Technique Selection: How Do Experts Do It? *11th IEEE International Requirements Engineering Conference (RE'03)*, Monterey Bay, USA. 169–178.

References – 5

J. Horkoff, E. Yu (2010). Interactive Analysis of Agent-Goal Models in Enterprise Modeling. *International Journal of Information Systems Modeling and Design* **1**(4):1–23.

IEEE (1990). *Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12-1990.

IEEE (1998). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Standard 830-1998.

ISO/IEC/IEEE (2017). *Systems and Software Engineering – Vocabulary*. ISO/IEC/IEEE Standard 24765.

ISO/IEC/IEEE (2018). *Systems and Software Engineering – Life Cycle Processes – Requirements Engineering*. ISO/IEC/IEEE Standard 29148.

ISO/IEC (2011). *Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models*. ISO/IEC Standard 25010:2011.

D. Jackson (2002). Alloy: A Lightweight Object Modelling Notation. *ACM Transactions on Software Engineering and Methodology* **11**(2):256–290.

M. Jackson (2005). Problem Frames and Software Engineering. *Information and Software Technology* **47**(14):903–912.

J. Jacky (1997). *The Way of Z: Practical Programming With Formal Methods*. Cambridge: Cambridge University Press.

Jacobson, I., M. Christerson, P. Jonsson, G. Övergaard (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Amsterdam; Reading, Mass.: Addison-Wesley.

N. Kano, S. Tsuji, N. Seraku, F. Takahashi (1984). Attractive Quality and Must-Be Quality. *Quality – The Journal of the Japanese Society for Quality Control* **14**(2):39–44.

References – 6

- A. van Lamsweerde (2001). Goal-Oriented Requirements Engineering: A Guided Tour. *5th IEEE International Symposium on Requirements Engineering (RE'01)*, Toronto, Canada. 249–261.
- A. van Lamsweerde (2004). Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice. *12th IEEE International Requirements Engineering Conference (RE'04)*, Kyoto, Japan. 4–7.
- Søren Lauesen (2002). *Software Requirements: Styles and Techniques*. Addison-Wesley.
- B. Lawrence, K. Wiegers and C. Ebert (2001). The Top Risks of Requirements Engineering. *IEEE Software* **18**(6):62–63.
- D. Leffingwell (2011). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Boston: Addison-Wesley.
- H. Lichter, M. Schneider-Hufschmidt, H. Züllighoven (1994). Prototyping in Industrial Software Projects – Bridging the Gap Between Theory and Practice. *IEEE Transactions on Software Engineering* **20**(11):825–832.
- P. Liggesmeyer (2002). *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. (in German) Berlin: Spektrum Akademischer Verlag.
- L. Macaulay (1993.) Requirements Capture as a Cooperative Activity. *First IEEE International Symposium on Requirements Engineering*, San Diego. 174–181.
- N. Maiden, A. Gizikis, S. Robertson (2004). Provoking Creativity: Imagine What Your Requirements Could Be Like. *IEEE Software* **21**(5):68–75.
- N. Maiden and S. Robertson (2005). Integrating Creativity into Requirements Processes: Experiences with an Air Traffic Management System. *13th IEEE International Requirements Engineering Conference (RE'05)*, Paris, France. 105–114.

References – 7

- A. Mavin, P. Wilkinson, A. Harwood, M. Novak (2009). Easy Approach to Requirements Syntax (EARS). *17th IEEE International Requirements Engineering Conference (RE'09)*, Atlanta, Georgia. 317–322.
- J. Mylopoulos (2006). *Goal-Oriented Requirements Engineering: Part II*. Slides of keynote talk at 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis, USA.
Available at: <https://files.ifi.uzh.ch/rerg/arvo/events/RE06/>
- B. Nuseibeh, J. Kramer, A. Finkelstein (2003). ViewPoints: Meaningful Relationships are Difficult! *25th International Conference on Software Engineering (ICSE'03)*, Portland, Oregon. 676–681.
- Object Management Group (2013). *Business Process Model and Notation (BPMN)*, version 2.0.2. OMG document formal/formal/2013-12-09 <https://www.omg.org/spec/BPMN>
- Object Management Group (2014). *Object Constraint Language*, version 2.4. OMG document formal/2014-02-03, <https://www.omg.org/spec/OCL/>
- Object Management Group (2017). *OMG Unified Modeling Language (OMG UML)*, version 2.5.1. OMG document formal/2017-12-05. <https://www.omg.org/spec/UML/>
- Object Management Group (2018). *OMG Systems Modeling Language (OMG SysML™)*, Version 1.6. OMG Document ptc/2018-12-08.
- P. Pepper, M. Broy, F. L. Bauer, H. Partsch, W. Dosch, M. Wirsing (1982). Abstrakte Datentypen: Die algebraische Spezifikation von Rechenstrukturen. (in German) *Informatik-Spektrum* 5(1):107–119.
- Klaus Pohl (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Berlin-Heidelberg: Springer.
- K. Pohl, Rupp, C. (2015). *Requirements Engineering Fundamentals*, 2nd edition. Rocky Nook. Distributed by O'Reilly Media, Sebastopol, CA.

References – 8

- D. G. Reinertsen (1997). *Managing the Design Factory – A Product Developer’s Toolkit*. The Free Press.
- D. G. Reinertsen (2009). *The Principles of Product Development Flow: Second Generation Lean Product Development*. Redondo Beach, Ca.: Celeritas Publishing.
- S. Robertson, J. Robertson (2012). *Mastering the Requirements Process*. 3rd edition. Boston: Addison-Wesley.
- S. Robertson, J. Robertson (eds.) *Volere Requirements Resources – Requirements Tools*.
<https://www.volere.org/requirements-tools/> Last visited 2019-09-12.
- C. Rupp et al. (2009). *Requirements Engineering und –Management* [in German]. 5th edition. Munich: Hanser.
- J. Ryser, M. Glinz (2001). Dependency Charts as a Means to Model Inter-Scenario Dependencies. In G. Engels, A. Oberweis and A. Zündorf (eds.): *Modellierung 2001. GI-Workshop*, Bad Lippspringe, Germany. GI-Edition - Lecture Notes in Informatics, Vol. P-1. 71–80.
- T.L. Saaty (1980). *The Analytic Hierarchy Process*. New York: McGraw Hill.
- A. Sillitti, G. Succi (2005). Requirements Engineering for Agile Methods. In A. Aurum, C. Wohlin (eds.) *Engineering and Managing Software Requirements*. Berlin: Springer. 309-326.
- I. Sommerville, P. Sawyer (1997). *Requirements Engineering: A Good Practice Guide*. Chichester: John Wiley & Sons
- J. M. Spivey (1992). *The Z Notation: A Reference Manual*. Second Edition. Hemel Hempstead: Prentice Hall International.
- A. Sutcliffe (1998). Scenario-Based Requirements Analysis. *Requirements Engineering* **3**(1):48–65.

References – 9

- W. Swartout, R. Balzer (1982). On the Inevitable Intertwining of Specification and Implementation. *Communications of the ACM* **25**(7):438–440.
- K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer (1998). Scenarios in System Development: Current Practice. *IEEE Software* **15**(2):34–45.
- K. Wieggers, J. Beatty (2013). *Software Requirements*. 3rd edition. Redmond, Wa.: Microsoft Press.
- J. B. Wordsworth (1992). *Software Development with Z: A Practical Approach to Formal Methods in Software Engineering*. Wokingham: Addison-Wesley.
- D. Wüest, N. Seyff, M. Glinz (2015). FlexiSketch Team: Collaborative Sketching and Notation Creation on the Fly. *37th International Conference on Software Engineering (ICSE 2015)*. 685–688.
- D. Wüest, N. Seyff, M. Glinz (2019). FlexiSketch: A Lightweight Sketching and Metamodeling Approach for End-Users. *Software and Systems Modeling* **18**(2):1513–1541.
- E. Yu (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE International Symposium on Requirements Engineering (RE'97)*. 226–235.
- P. Zave, M. Jackson (1997). Four Dark Corners of Requirements Engineering. *ACM Transactions on Software Methodology* **6**(1):1–30.
- D. Zowghi, C. Coulin (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In A. Aurum, C. Wohlin (eds.) *Engineering and Managing Software Requirements*. Berlin: Springer. 19–46.