Part I: The Fundamentals

Part II: Requirements Engineering Practices

Part III: Enablers and Stumble 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 (2013). *A Glossary of Requirements Engineering Terminology*, Version 1.5. International Requirements Engineering Board (IREB). Originally published in 2011.
Available at http://www.ireb.org (check-out 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. Robertson, J. Robertson (2006). *Mastering the Requirements Process*. 2nd edition. Boston: Addison-Wesley.

K. Pohl (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques.* Berlin: Springer.

K. Pohl, Rupp, C. (2011). *Requirements Engineering Fundamentals.* Rocky Nook. Distributed by O'Reilly Media, Sebastopol, CA.

K. Wiegers (2006). *More About Software Requirements: Thorny Issues and Practical Advice*. Redmond: Microsoft Press.

References

Chapter 1

D.C. Gause, G.M. Weinberg (1989). *Exploring Requirements: Quality before Design*. New York: Dorset House.

M. Glinz (2004). *Software Engineering I* [in German]. Lecture notes, University of Zurich.

M. Glinz, R. Wieringa (2007). Stakeholders in Requirements Engineering. *IEEE Software* **24**(2). 18–20.

M. Glinz (2013). *A Glossary of Requirements Engineering Terminology*, Version 1.5. International Requirements Engineering Board (IREB). Originally published in 2011.
Available at http://www.ireb.org (check-out CPRE Glossary)

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

ISO/IEC/IEEE (2011). *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*. ISO/IEC/IEEE Standard 29148, First edition 2011-12-01

L. Macaulay (1993.) Requirements Capture as a Cooperative Activity. *Proceedings 1st IEEE International Symposium on Requirements Engineering*, San Diego. 174–181.

B. Nuseibeh, J. Kramer, A. Finkelstein (2003). ViewPoints: Meaningful Relationships are Difficult! *Proceedings 25th International Conference on Software Engineering (ICSE'03)*, Portland, Oregon. 676–681.

References – 2

Chapter 2

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

P. Zave, M. Jackson (1997). Four Dark Corners of RequirementsEngineering. *ACM Transactions on Software Methodology* **6**(1). 1–30.

W. Swartout, R. Balzer (1982). On the Inevitable Intertwining of Specification and Implementation. *Communications of the ACM* **25**(7). 438–440.

Chapter 3

M. Glinz (2005). Rethinking the Notion of Non-Functional Requirements. *Proceedings of the Third World Congress for Software Quality (3WCSQ 2005)*, München, Vol. II. 55–64.

M. Glinz (2007). On Non-Functional Requirements. *Proceedings 15th IEEE International Requirements Engineering Conference*, Delhi, India. 21–26.

Chapter 4

M. Glinz, S.A. Fricker (2014). On Shared Understanding in Software Engineering: an Essay. Computer Science – Research and Development 1-14} Computer Science – Research and Development. DOI 10.1007/s00450-014-0256-x Published online 1 July 2014.

References – 3

Chapter 5

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

S. Robertson, J. Robertson (2006). *Mastering the Requirements Process*. 2nd edition. Boston: Addison-Wesley.

Chapter 6

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.

Chapter 7

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.

O. Dieste, N. Juristo, F. Shull (2008). Understanding the Customer: What Do We Know about Requirements Elicitation? *IEEE Software* **25**(2). 11–13.

D. C. Gause, G.M. Weinberg (1989). *Exploring Requirements: Quality before Design*. New York: Dorset House.

M. Glinz, R. Wieringa (2007). Stakeholders in Requirements Engineering. *IEEE Software* **24**(2). 18–20.

References – 4

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. *Proceedings GI Conference Software Engineering 2013, Lecture Notes in Informatics vol P-213*. 19–35.

J. Goguen, C. Linde (1993). Techniques for Requirements Elicitation. *Proceedings of the First IEEE International Symposium on Requirements Engineering (RE'93)* San Diego, California, USA. 152–164.

E. Gottesdiener (2002). *Requirements by Collaboration: Workshops for Defining Needs*. Boston: Addison-Wesley.

A.M. Hickey, A.M. Davis (2003). Elicitation Technique Selection: How Do Experts Do It? *Proceedings 11th IEEE International Requirements Engineering Conference (RE'03)*, Monterey Bay, USA. 169–178.

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.

A. van Lamsweerde (2001). Goal-Oriented Requirements Engineering: A Guided Tour. *Proceedings 5th IEEE International Symposium on Requirements Engineering (RE'01)*, Toronto, Canada. 249–261.

S. Robertson, J. Robertson (2006). *Mastering the Requirements Process*. 2nd edition, Boston: Addison-Wesley.

E. Yu (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings 3rd IEEE International Symposium on Requirements Engineering (RE'97)*. 226–235.

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.

References – 5

Chapter 8

I. Alexander, R. Stevens (2002). *Writing Better Requirements*. London: Addison-Wesley.

M. Cohn (2004). *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley.

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.

C. Rupp et al. (2009). *Requirements Engineering und –Management* [in German]. 5th edition. Munich: Hanser.

Chapter 9

G. Booch (1986). Object-Oriented Development. *IEEE Transactions on Software Engineering* **12**(2). 211–221.

G. Booch (1994). *Object Oriented Analysis and Design with Applications*. Second Edition. Redwood City, Ca.: Benjamin/Cummings.

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.

T. DeMarco (1978). *Structured Analysis and System Specification*. New York: Yourdon Press.

References – 6

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. *Proceedings of the Second World Congress on Software Quality*. Yokohama. 55–60.

M. Glinz (2000b). Problems and Deficiencies of UML as a Requirements Specification Language. *Proceedings of the 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 (2013). *A Glossary of Requirements Engineering Terminology*, Version 1.5. International Requirements Engineering Board (IREB). Originally published in 2011.
Available at http://www.ireb.org (check-out CPRE Glossary)

D. Harel (1988). On Visual Formalisms. *Communications of the ACM* **31**(5). 514–530.

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.

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

A. van Lamsweerde (2001). Goal-Oriented Requirements Engineering: A Guided Tour. *Proceedings 5th IEEE International Symposium on Requirements Engineering (RE'01)*, Toronto, Canada. 249–261.

References – 7

A. van Lamsweerde (2004). Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice. *Proceedings 12th IEEE International Requirements Engineering Conference (RE'04)*, Kyoto, Japan. 4–7.

J. Mylopoulos (2006). *Goal-Oriented Requirements Engineering: Part II*. Slides of keynote talk at 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis, USA. http://www.re06.org

Object Management Group (2011a). *Business Process Model and Notation (BPMN)*, version 2.0. OMG document formal/2011-01-03. http://www.omg.org/spec/BPMN/2.0/PDF

Object Management Group (2011b). *Unified Modeling Language: Superstructure,* version 2.4.1. OMG document formal/2011-08-06. http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF

Object Management Group (2012). *Object Constraint Language, v2.3.1* OMG document formal/2012-05-09 http://www.omg.org/spec/OCL/2.3.1/PDF

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.

A. Sutcliffe (1998). Scenario-Based Requirements Analysis. *Requirements Engineering Journal* **3**(1). 48–65.

K. Weidenhaupt, K. Pohl, M. Jarke, P. Haumer (1998). Scenarios in System Development: Current Practice. *IEEE Software* **15**(2). 34–45.

E. Yu (1997). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings 3rd IEEE International Symposium on Requirements Engineering (RE'97)*. 226–235.

References – 8

Chapter 10

J.-R. Abrial (2009). *Modelling in Event-B: System and Software Engineering*. Cambridge: Cambridge University Press.

D. Björner, C. Jones (1978). *The Vienna Development Method*. Berlin: Springer.

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

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

Object Management Group (2012). *Object Constraint Language, v2.3.1* OMG document formal/2012-05-09 http://www.omg.org/spec/OCL/2.3.1/PDF

Object Management Group (2011b). *Unified Modeling Language: Superstructure,* version 2.4.1. OMG document formal/2011-08-06. http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF

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.

J.M. Spivey (1992). *The Z Notation: A Reference Manual*. Second Edition. Hemel Hempstead: Prentice Hall International.

J.B. Wordsworth (1992). *Software Development with Z: A Practical Approach to Formal Methods in Software Engineering*. Wokingham: Addison-Wesley.

References – 9

Chapter 11

B. Boehm (1981). *Software Engineering Economics*. Englewood Cliffs: Prentice Hall.

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.

P. Liggesmeyer (2002). *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. (in German) Berlin: Spektrum Akademischer Verlag.

K. Pohl, Rupp, C. (2011). *Requirements Engineering Fundamentals.* Rocky Nook. Distributed by O'Reilly Media, Sebastopol, CA.

Chapter 12

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.

N. Maiden, A. Gizikis, S. Robertson (2004). Provoking Creativity: Imagine What Your Requirements Could Be Like. *IEEE Software* **21**(5) (Sept./Oct. 2005). 68–75.

N. Maiden and S. Robertson (2005). Integrating Creativity into Requirements Processes: Experiences with an Air Traffic Management System. Proceedings *13th IEEE International Requirements Engineering Conference (RE'05)*, Paris, France. 105–114.

References – 10

Chapter 13

J. Dick (2005). Design Traceability. *IEEE Software* **22**(6). 14–16.

O. Gotel, A. Finkelstein (1994) An Analysis of the Requirements Traceability Problem, *Proceedings 1st International Conference on Requirements Engineering,* Colorado Springs. 94–101.

K. Pohl, Rupp, C. (2011). *Requirements Engineering Fundamentals.* Rocky Nook. Distributed by O'Reilly Media, Sebastopol, CA.

T.L. Saaty (1980). *The Analytic Hierarchy Process.* New York: McGraw Hill.

K.E. Wiegers (1999). *Software Requirements*. Redmond, WA: Microsoft Press.

Chapter 14

S. Robertson, J. Robertson (eds.) *Requirements Tools*. http://www.volere.co.uk/tools.htm. Last visited 2013-10-06.