1 Introduction to Requirements Engineering

The first lecture provides an overview on the domain of requirements engineering and software specification. In particular, you learn about the history and importance of requirements engineering (RE); the role of RE within software engineering and its connections with other disciplines; understand the RE process and its phases; and we will discuss different types of requirements, as well as common errors and defects when writing requirements. Finally, the importance of requirements engineering for software product development and its impact on costs and schedules is discussed. Common obstacles to good RE practise are highlighted and agile methodologies are introduced as one way of coping with these obstacles. The detailed methodologies for each of these steps will be the content of the following lectures. 1.1

Learning Objectives

Learning Objective 1:

- Understand the role of Requirements Engineering (RE) as a discipline
- Be able to define the RE process
- Understand the connection between RE and other software lifecycle activities
- Learn the connections between RE and other disciplines
- Understand the purposes of domain understanding, requirements elicitation, requirements evaluation and agreement, requirements specification and documentation, and requirements validation and verification

Learning Objective 2:

- Understand the difference between descriptive and prescriptive requirements
- Understand the difference between system and software requirements
- Learn the distinction between domain properties, assumptions, and definitions
- Learn the categorization of functional vs. non-functional requirements
- Understand the target qualities for requirements
- Learn about common defects and errors in requirements documents
- Understand the different types of projects (greenfield vs. brownfield, customer-driven vs. market-driven) and their impact on the RE process

Learning Objective 3:

- Understand the importance of RE for software projects
- Learn about obstacles to good RE practise in projects
- Learn about the impact of RE on a project's cost, schedule, and overall success
- Understand the ideas of agile software development processes and their connection with RE

Notes and Further Reading

Read [4, Chapter 1] for a general introduction to requirements engineering. This chapter also introduces a number of case studies that we will use throughout the course. It is highly recommended that you also read [2, Chapters 1, 2]. For further reading, Pohl's reference book provides another good overview [3, Chapters 1, 2].

While the main focus of this course is on the requirements part of software engineering, we will look at its connection to the (object-oriented) design and implementation phase, using Larman's seminal textbook [1]. If you are not familiar with it, it is recommended you start by reading [1, Chapter 1].

Reading Material

Required

- [4, Chapter 1] (Setting the Scene)

Supplemental

- [2, Foreword, Chapters 1, 2] (Introduction to RE)
- [1, Chapter 1] (Applying UML and Patterns Introduction)

Further Reading

- [3, Chapters 1, 2] (Motivation, Requirements)

1.4

Bibliography

[1] Craig Larman. *Applying UML and Patterns*. Prentice Hall, third edition, 2005.

[2] Dean Leffingwell and Don Widrig. *Managing Software Requirements: A Use Case Approach*. Addison-Wesley, 2003. Available online at http://clues.concordia.ca/record=b2529323.

[3] Klaus Pohl. *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer-Verlag Berlin Heidelberg, 2010.

[4] Axel van Lamsweerde. *Requirements Engineering: From System Goals to UML Models to Software Specifications*. John Wiley & Sons, 2009.

Study Questions

1. Why is requirements engineering important?
2. What are the main findings of the 1995 “Chaos Report” by the Standish Group?
3. What are the main reasons for failed or challenged projects? How can requirements engineering help prevent this from occurring?
4. What are the main obstacles to good requirements engineering?
5. What is the difference between functional (FR) and non-functional (NFR) requirements?
6. What are requirements taxonomies and why are they helpful?
7. What is the importance of NFR for software development?
8. Name and define 5 target qualities for software requirements specifications.
9. What is the difference between descriptive and prescriptive requirements statements, and why is this important for RE?
10. What are quantifiable requirements?
11. What are the “Why”, “Who” and “What” dimensions of RE?
12. What are greenfield and brownfield projects? What is the impact of the project type for requirements engineering?
13. What are customer-driven vs. market-driven projects and their impact on RE?
14. What are stakeholders?
15. Define “problem world”, “machine solution”, “system-as-is” and “system-to-be”
16. Explain the differences between “System requirement” vs. “Software requirement”
17. Explain the differences between “Domain property”, “Assumption” and “Definition”
18. How can you define the RE process? Name the individual steps, their input and the produced artifacts.