
CHAPTER 10

Documenting the requirements

- This chapter addresses the purpose, structure, and contents of the SRS (Software Requirement Specification)
- After finish this chapter, student should understand the purpose, structure of the SRS.
- Student could understand what they do and what they have to write in every section of the SRS document.
- Student could enhance the difference in SRS document between Agile projects and other projects

1. Why is documenting the requirements?
2. The software requirements specification
3. A software requirements specification template
4. Requirements specification on agile projects

Why is documenting the requirements?

- The result of requirements development is a documented agreement among stakeholders about the product to be built.
- The product's functional and nonfunctional requirements often are stored in a software requirements specification, or SRS, which is delivered to those who must design, build, and verify the solution.
- You can represent software requirements in several ways, including:
 - Well-structured and carefully written natural language.
 - Visual models that illustrate transformational processes, system states and changes between them, data relationships, logic flows, and the like.
 - Formal specifications that define requirements by using mathematically precise specification languages.

The software requirements specification

- It is sometimes called a business requirements document (BRD), functional specification, product specification, system specification, or simply requirements document.
- Audiences rely on the SRS: Customers, Project managers , Software development teams , Testers, Maintenance and support staff, Documentation writers , Training personnel , Legal staff , Subcontractors

The software requirements specification

- How many specifications?
- How to organize and write SRS (detail in page 185)
- Purpose, how to do?
 - Labeling requirements
 - Sequence number
 - Hierarchical numbering
 - Hierarchical textual tags
- Dealing with incompleteness
- User interfaces and the SRS

A software requirements specification template

1. Introduction

- 1.1 Purpose
- 1.2 Document conventions
- 1.3 Project scope
- 1.4 References

2. Overall description

- 2.1 Product perspective
- 2.2 User classes and characteristics
- 2.3 Operating environment
- 2.4 Design and implementation constraints
- 2.5 Assumptions and dependencies

3. System features

- 3.x System feature X
 - 3.x.1 Description
 - 3.x.2 Functional requirements

4. Data requirements

- 4.1 Logical data model
- 4.2 Data dictionary
- 4.3 Reports
- 4.4 Data acquisition, integrity, retention, and disposal

5. External interface requirements

- 5.1 User interfaces
- 5.2 Software interfaces
- 5.3 Hardware interfaces
- 5.4 Communications interfaces

6. Quality attributes

- 6.1 Usability
- 6.2 Performance
- 6.3 Security
- 6.4 Safety
- 6.x [others]

7. Internationalization and localization requirements

8. Other requirements

Appendix A: Glossary

Appendix B: Analysis models

Requirements specification on agile projects

- The extent to which just-in-time informal verbal and visual communication between customers and developers can supply the necessary details to permit the correct implementation of each user requirement
- The extent to which informal communication methods can keep the team effectively synchronized across time and space
- The extent to which it is valuable or necessary to retain requirements knowledge for future enhancement, maintenance, application reengineering, verification, statutory and audit mandates, product certification, or contractual satisfaction
- The extent to which acceptance tests can serve as effective replacements for descriptions of the expected system capabilities and behaviors
- The extent to which human memories can replace written representations