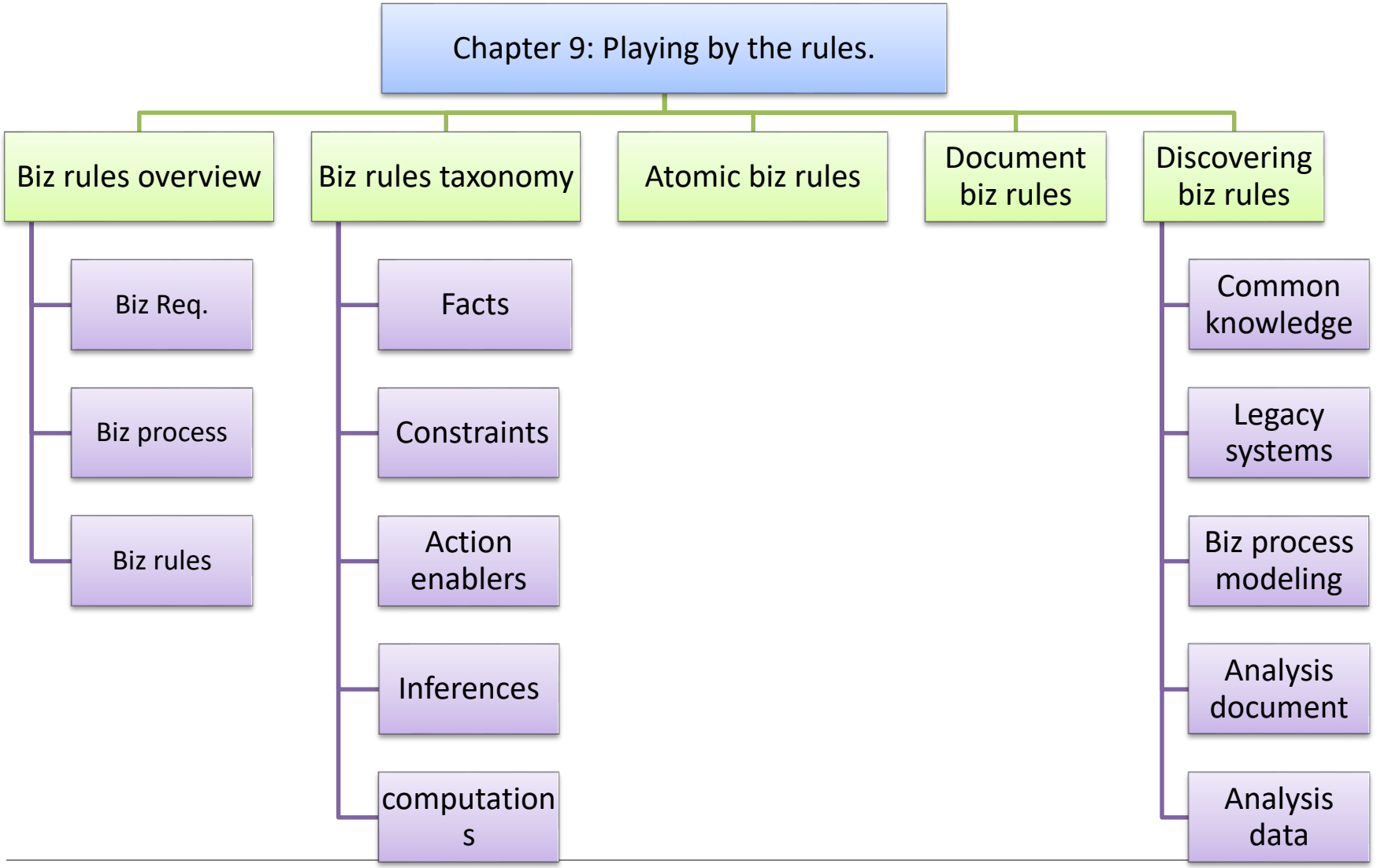
FPT UNIVERSITY

ĐẠI HỌC FPT CẦN THƠ

CHAPTER 10

Documenting the requirements

Review chapter 9



Objectives

- 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.

Contents

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

- The SW requirements specification (SRS) is sometimes called a *business requirements document* (BRD), *functional specification*, *product specification*, *system specification*, or simply *requirements document*.
- The SRS states the functions and capabilities that a SW system must provide, its characters, and the constraints that it must respect.
- The SRS is the basis for subsequent project planning, design, and coding, as well as the foundation for system testing and user documentation. However, it should not contain design, construction, testing or project management details other than known design and implementation constraints.

The software requirements specification

- Audiences rely on the SRS:
  - Customers, the marketing department, and sales staff need to know what product they can expect to be delivered.
  - Project managers base their estimate of schedule, effort, and resources on requirements.
  - Software development teams need to know what to build.
  - Testers use it to develop requirements-based tests, test plans, and test procedures.
  - Maintenance and support staff use it to understand what each part of the product is supposed to do.
  - Documentation writers base user manuals and help screens on SRS and the user interface design.
  - Training personnel use it and user documentation to develop educational materials.
  - Legal staff ensures that the requirements comply with applicable laws and regulations.
  - Subcontractors base their work on – and can be legally held to – the specified requirements.

The software requirements specification

- How many specifications? Most projects will create just one SW requirements specification. This is not practical for large projects. Large systems projects often write a system requirements specification, followed by separate SW and perhaps HW requirements specifications.
- How to organize and write SRS (detail in page 185)
  - Use an appropriate template to organize all the necessary information.
  - Label and style sections, subsections, and individual requirements consistently.
  - Use visual emphasis (bold, italic, underline, color…)
  - Create a table of contents to read clearly.
  - Number all figures and tables, give them captions, and refer to them by number.
  - If you are using documents, define hyperlinks to let the reader navigate to related information.
  - Include visual representations of information when possible to facilitate understanding.
  - …

The software requirements specification

- Purpose, how to do?
  - Labeling requirements: Every requirement needs a unique and persistent identifier. This allow you to refer to specific requirements in a change request, modification history, cross-reference, or requirements traceability matrix.
  - Sequence number: The simplest approach gives every requirement a unique sequence number, such as UC-9 or FR-26. Commercial requirements management tools assign such an identifier when a user adds a new requirement to the tool's database.
  - Hierarchical numbering: In the most commonly used convention, if the functional requirements appear in section 3.2 of your SRS, they will all have labels that begin with 3.2.
  - Hierarchical textual tags: are structured, meaningful, and unaffected by adding, deleting, or moving other requirements. *EX: Product.Cart; Product.Discount.Error; Product.Shipping;….*

The software requirements specification

- Dealing with incompleteness: Sometimes you know that you lack a piece of information about specific requirement. Use the notation TBD (to be determined) to flag these knowledge gaps. Plan to resolve all TBDs before implementing a set of requirements. *Record TBDs and other requirements questions in an issues list to be resolved in the near future.*
- User interfaces and the SRS: Incorporating user interface designs in the SRS has both benefits and drawbacks.
  - On the plus side, user interfaces with paper prototypes, working mock-ups, wireframes, or simulation tools makes requirements tangible to both users and developers.
  - On the negative side, screen images and user interface architectures describe solutions and might not truly be requirements. User interfaces don't replace written user and functional requirements.

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

Read from page 190 to page 198 for more information.

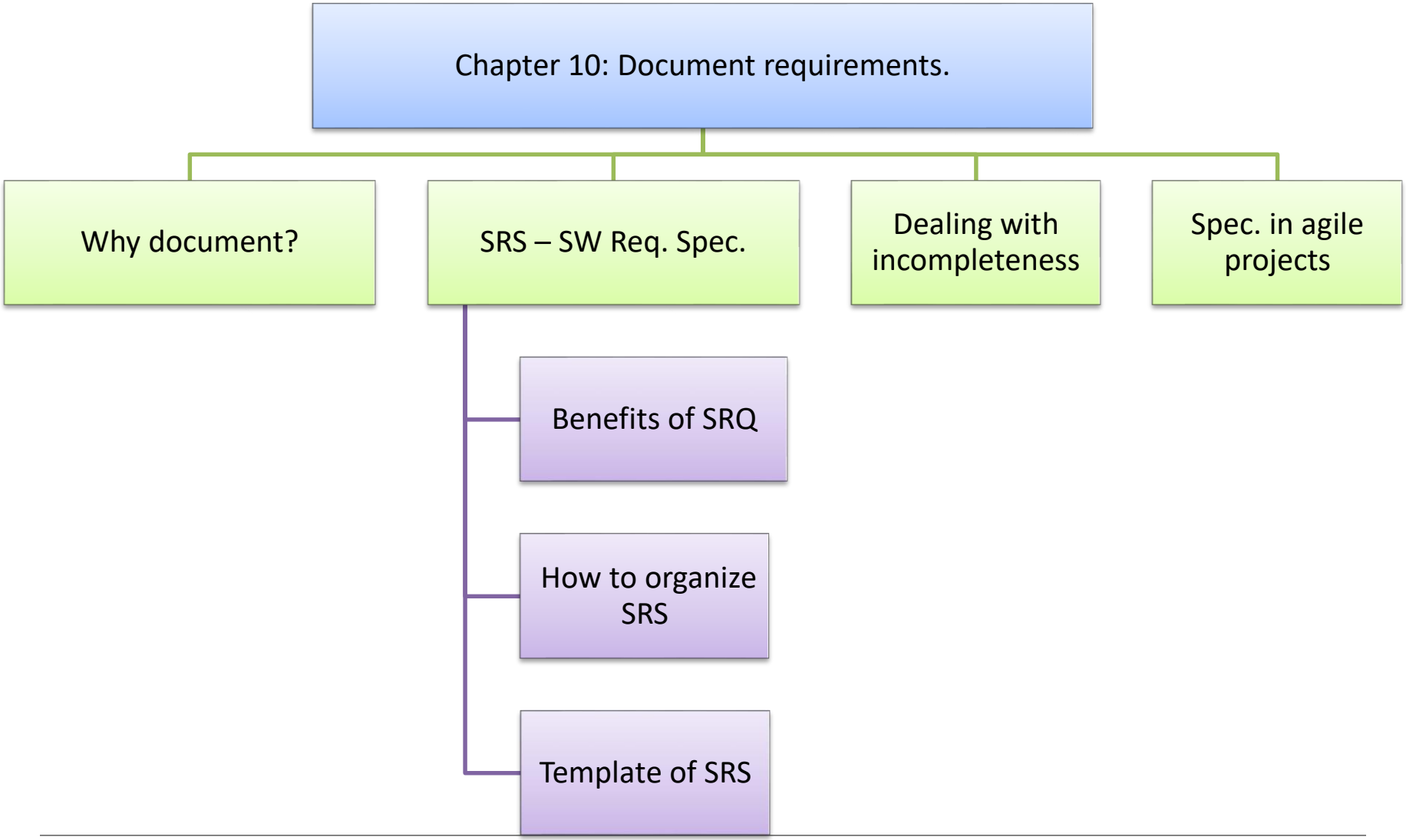
Requirements specification on agile projects

Remember the common goal of requirements development: to accumulate a shared understanding of requirements that is *good enough* to allow construction of the next portion of the product to proceed at an acceptable level risk.

In agile projects, to document requirements depends on factors including the following:

- 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 (ủy quyền theo luật) and audit mandates (kiểm toán), product certification (chứng nhận sản phẩn), 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.

Review chapter 10



THE END

THANK YOU!