Requirements and Specification, ESOF 328, Spring 2022
Review for Exam 1, Feb. 4

Part I: Software requirements: What, Why, and Who

The essential software requirement, Chapter 1

- Be able to define what is meant by software requirements
- Be able to categorize information into the various types of requirements:

Term
Business requirements
Business rule
Constraint
External interface requirement
Feature
Functional requirement
Nonfunctional requirement
Quality attribute
User requirement

Table 1.1, page 7 (also Figure 7.7, page 135)

- Know the relation of business requirements, user requirements and functional requirements (Figure 1-3, page 12)
- Know the difference between product and project requirements
- Know the sub-disciplines of requirements: elicitation, analysis, specification, validation and management (Figure 1-4, page 15, details on development portion in Figure 3-1, page 45)
- Given an activity, be able to identify in what phase it belongs
- Know the common problems of requirements (pages 20-22)

Requirements for customer perspective, Chapter 2

- Know possible stakeholders (Figure 2-2, page 28)
- Know Wiegers recommendations for the signoff-off process and document

Good practices for requirements engineering, Chapter 3

- Know the four main stages in the requirements development process (Figure 3-1, page 45, first four sub-disciplines). These are referred to as sub-disciplines in Chapter 1.
- Identify, and be able to describe, at least three practices in each of the stages above which you expect to be the most valuable for our project (Table 3-1, page 44, expanded on in pages 48-57)

The business analyst, Chapter 4

- Know the tasks business analyst do
- Know the skills business analysts need

Part II: Requirements Development

Establishing the business requirements, Chapter 5

- Know the purpose of a vision statement and be able to write one
- Know ways to visual scope: Context diagram (pg. 93), Ecosystem map (pg. 94), Feature tree (pg. 95), Event list (pg. 96)

Finding the voice of the user, Chapter 6

- Know a variety of sources from which requirements can be gotten
- Know the purpose and importance of a product champion
- Know the importance of talking to the right people

Hearing the Voice of the Customer, Chapter 7

- Know ways to solicit information from users
- Be able to categorize information that is gotten from the customer, see Figure 7.7, page 135

Understanding user requirements, Chapter 8

- Know the purpose and ways in which use cases, casual and "fully dressed" can be used
- Be able to create a use case diagram
- Know the difference between a context diagram and a use case diagram
- Be able to write "fully dressed" use cases
- Know how to use the "extends" and "includes" relationships between use cases
- Identify, and be able to describe, at least three use case traps best avoided in our project.

Documenting the requirements, Chapter 10

- Know the 5 sections of the Montana Tech SRS template and what is in each
- Given a subsection of the Montana Tech SRS template, be able to describe what belongs in the section
- Be able to give guidelines for writing good requirements
- Be able to write realistic, unambiguous, consistent, complete, testable, independent, clear requirements which use the active voice, avoid jargon, avoid premature design, and can't be broken into multiple more clear requirements
- Terms shall, should and will (page 209, Chapter 11)
    - shall – requirement, desired functionality, system capability (some use "must", "needs to", "has to") – can see this as imperative
    - should – desired (some say "may", "could")
    - will – design expectation – can see this as declarative (something that is true but that developers do not need to implement

A Picture is Worth 1024 Words, Chapter 12

- Be able to describe the purpose of each of the following:
    - Context diagram (DFD Level 0)
    - Ecosystem Map
    - Feature Tree
    - Use-case diagram (UML)
    - Use-case
    - Activity (UML)/Object flow diagram
    - Data flow diagram
    - Swimlane diagram
    - State transition diagram
    - Dialog map
- Be able to suggest a diagram(s) which may help communicate about and think about a situation, and be able to support your suggestion.
- Be able to describe situations in which a diagram is likely to help think about the situation and communicate information about it.
- Be able to create:
    - Context diagram (DFD Level 0)
    - Ecosystem Map
    - Feature Tree
    - Use-case diagram (UML)
    - Use-case
    - Activity (UML)
    - Data flow diagram
    - Swimlane diagram
    - State transition diagram
    - Dialog map