Requirements and Specification, ESOF 328, Spring 2021
Exam 2, March 19

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

Specifying data requirements, Chapter 13

- Know that a data model provides a high level view of the data while a data dictionary provides a low level view of the data
- Know that data dictionary entries include names, meaning, composition or data types, and allowed values
- CRUD – Create, Read, Update and Delete, basic operations which are needed for most objects. Some add L for list.
- CRUD matrix is useful for identifying missing requirements (Figure 13.5, page 252). Place the objects/entities going across and the use cases going down. In the cell say if the use case creates the object, reads it, updates it, deletes it or lists it. Easy to see if one of these items is missing.
- Know that specifying the content and format of the reports needed is an important aspect.

Risk reduction through prototyping, Chapter 15

- Know that prototyping can be used to clarify/validate requirements, explore design alternatives, and/or to create a subset that will grow into the ultimate product.
- Know the importance of being clear about why you are creating the prototype, what you expect to learn from the prototype and what you'll do with the prototype when it is done
- Know the risks and advantages of creating prototypes

First things first: Setting requirement priorities, Chapter 16

- Know the importance of prioritizing requirements: deliver maximum business value as quickly as possible, provide highest value at lowest cost, because you can't do it all
- Know that prioritization needs to occur at some level – features, use cases or user stories, or functional requirements
- Know issues of prioritization – relative importance to the customers, timing at which capabilities needs to be delivered, requirements that serve as predecessors for other requirements, the cost to satisfy a requirement

Validating the requirements, Chapter 17

- Know the V-model of software development (starting test planning and test-case development in parallel with requirements development, you'll detect many errors shortly after they're introduced , page 330)
- Validating versus verification
    - Validation – are we building the right system
    - Verification – are we building the system right

Beyond requirements development, Chapter 19

- Know the guideline: approximately 15% of total project effort should be devoted to requirements work
- Know what is meant by analysis paralysis
- Be able to discuss the balance between not including user interface design, yet, "user interface design is so closely related to requirements that it shouldn't be pushed downstream to be done without end-user engagement"

Part III: Requirements for Specific Project Classes

Agile projects, Chapter 20

- Be able to characterize the difference between agile and non-agile development
- Know the difference between "user story", "use case", "epic", "feature"
- Be able to describe the agile process
- Know the idea of Minimum marketable feature (MMF) for use stories

Part IV: Requirements Management

Requirements management practices, Chapter 27

- Know the major requirements management activities and what is involved in each (Figure 27-1, page 458)
- Know what a requirements baseline is

Change Happens, Chapter 28

- Know the process for a change request suggested in the text, page 447
- Be able to describe the purpose and composition of a Change Control Board

Links in the Requirements Chain, Chapter 29

- Know about tracing requirements and be able to give example elements for the "forward to", "backwards from", "forward from" and "backward to" links.

Tools for requirements engineering, Chapter 30

- Know that tools cannot compensate for a lack of business analysis and requirements engineering process, training, discipline or experience
- Identify, and be able to describe, at least two tools which you expect to be useful for our project, for each of the stages: elicitation, prototyping, and modeling

Risk management, Chapter 32

- Know what a risk is, how it differs from an issue, and to express risks as condition-consequence
- Know the elements of risk management – assessment, avoidance and control, and what is involved in each
- Know what is meant by risk exposure
- Know that risks need to be monitored