Requirements Engineering & Software Specification Document

Ohjelmankehityspr., versionhallinta ja testaus – Chapter 10

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Deepak K.C. ; deepak.kc@kamk.fi

Requirements engineering

Process of understanding

- customer requirements
- Constraints under which software operates and is developed

Requirements

- High level abstract statement of a system
- Detailed mathematical functional specification

Types of requirements

- User requirements
  - Satements in natural language
  - Diagrams of the services the system should provide
- System requirements
  - Structured document
  - Detailed description of the system’s functions, services & operational constrains
  - Defines what should be implemented

User & System requirements

User requirement definition

1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20 mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

Functional & non-functional requirements

Functional requirements

- What system should do?
- How system should react to particular inputs?
- How the system should behave in particular situations?

Non-functional requirements

- Constrains on the services or functions offered by the system
- Example: timing constraints, standards, constraints on the development process
- Often applies to the system as a whole rather than individual features or services

Example Function requirements for the MHC-PMS

- A user shall be able to search the appointments lists for all clinics.
- The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Example: Non function requirements for the MHC-PMS

Product requirement
The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement
Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.

External requirement
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

Usability requirements

- System should be easy to use
- User errors are minimized
- Easy to learn
- Easy to remember

Software requirements document

- Official statement of what is required
- Must include user requirements and a specification of the system requirements
- Not a design document
- Should focus more on what the system should do rather than how it should do

Agile methods & requirements

- Agile methods argue that producing a requirements document is a waste of time due to rapidly changing requirements
- XP uses incremental requirements and express requirements as user stories
- Practical for business systems but problematic for critical systems or systems developed by several teams

Software requirements specification document

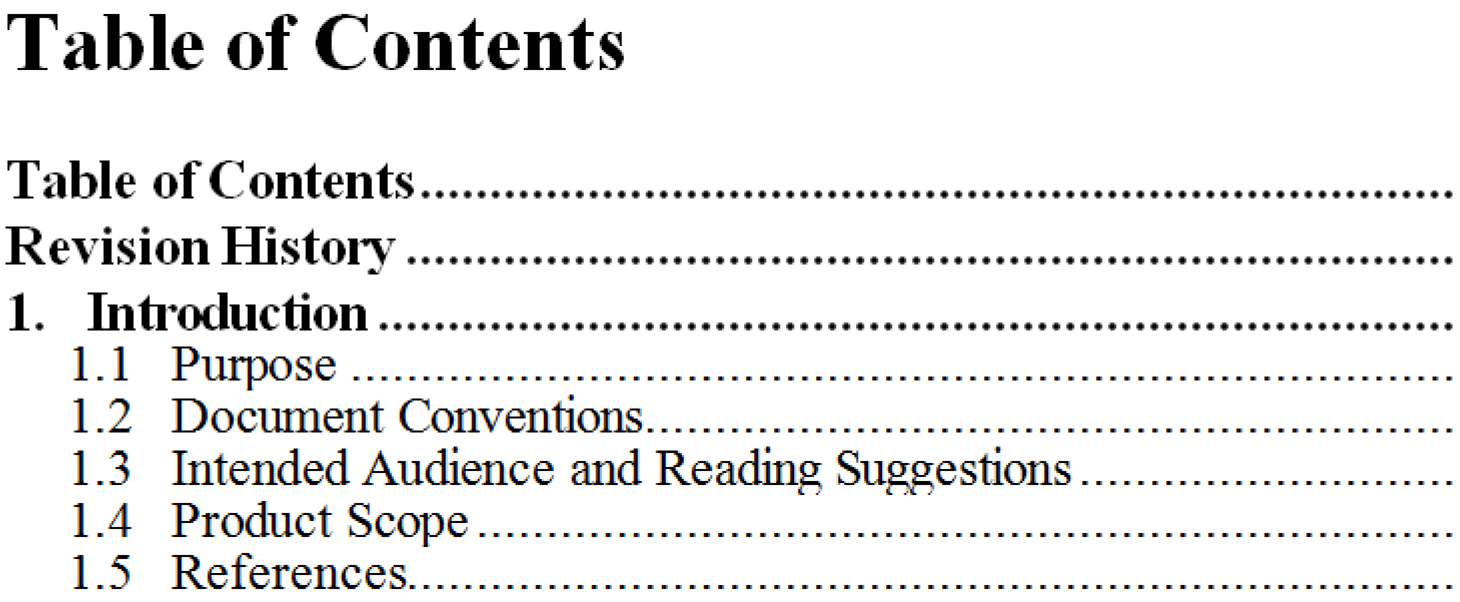
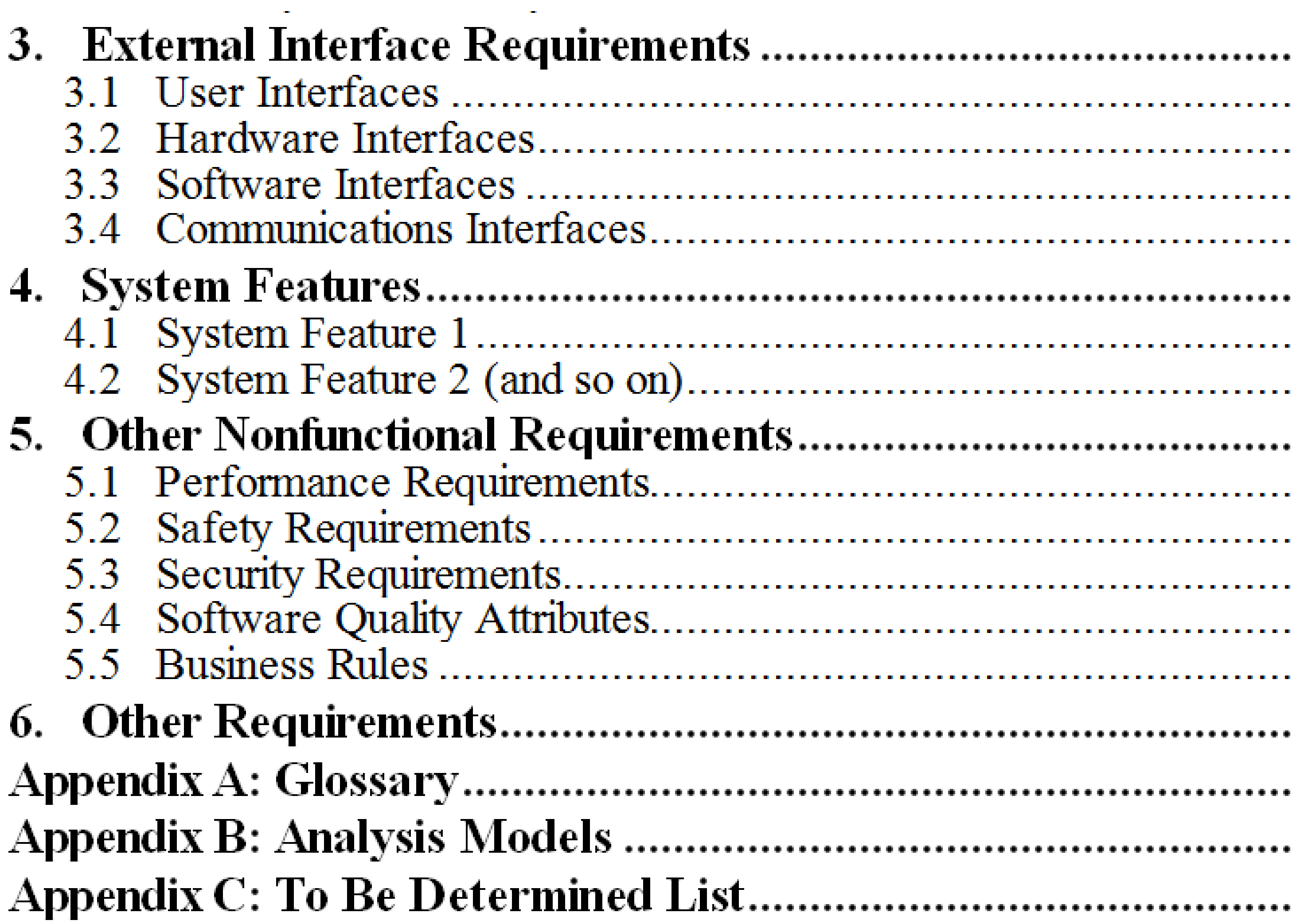
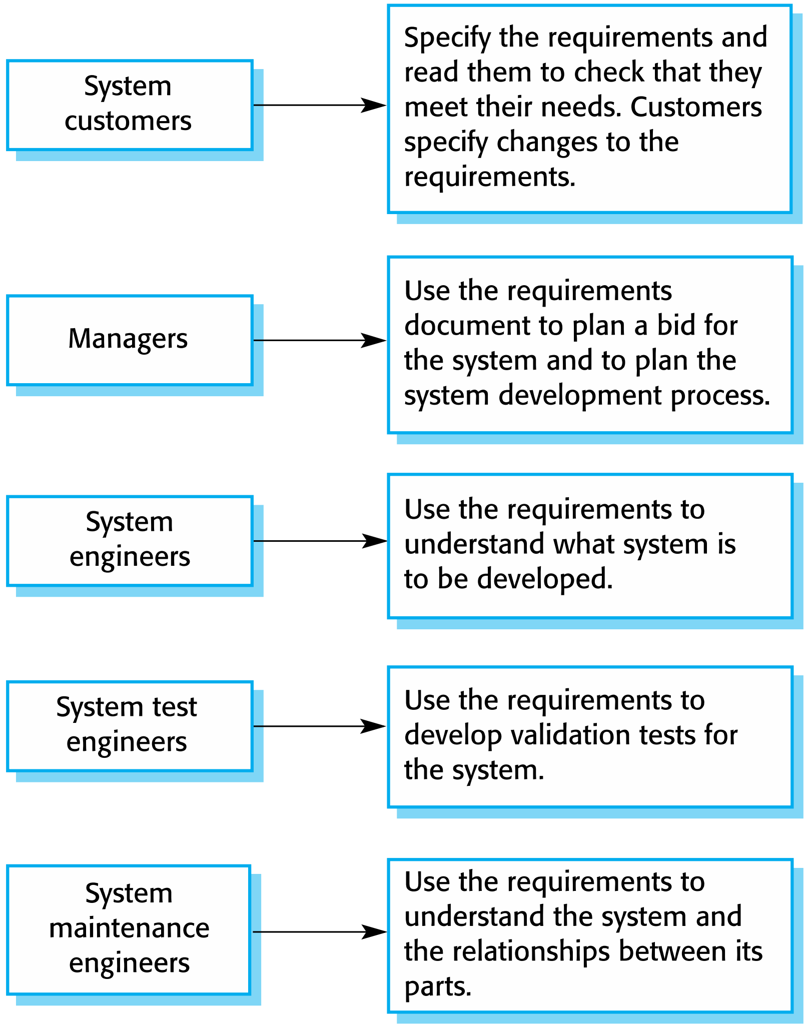


Table of Contents



Users of requirement document


System customers
Specify the requirements and read them to check that they meet their needs. Customers specify changes to the requirements.
Managers
Use the requirements document to plan a bid for the system and to plan the system development process.
System engineers
Use the requirements to understand what system is to be developed.
System test engineers
Use the requirements to develop validation tests for the system.
System maintenance engineers
Use the requirements to understand the system and the relationships between its parts.

Requirements document variability

- Information in requirements document depends on type of system and the approach to development used.
- Systems developed incrementally will, typically, have less detail in the requirements document.
- Requirements documents standards have been designed e.g. IEEE standard. These are mostly applicable to the requirements for large systems engineering projects.

Requirements specification

- User requirements must be understandable to end users and customers who do not have a technical background
- System requirements are more details requirements and include more technical information
- Requirement document can be part of a contract for the system development

Guidelines for writing requirements

- Invent a standard format and use it for all requirements.
- Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon.
- Include an explanation (rationale) of why a requirement is necessary.

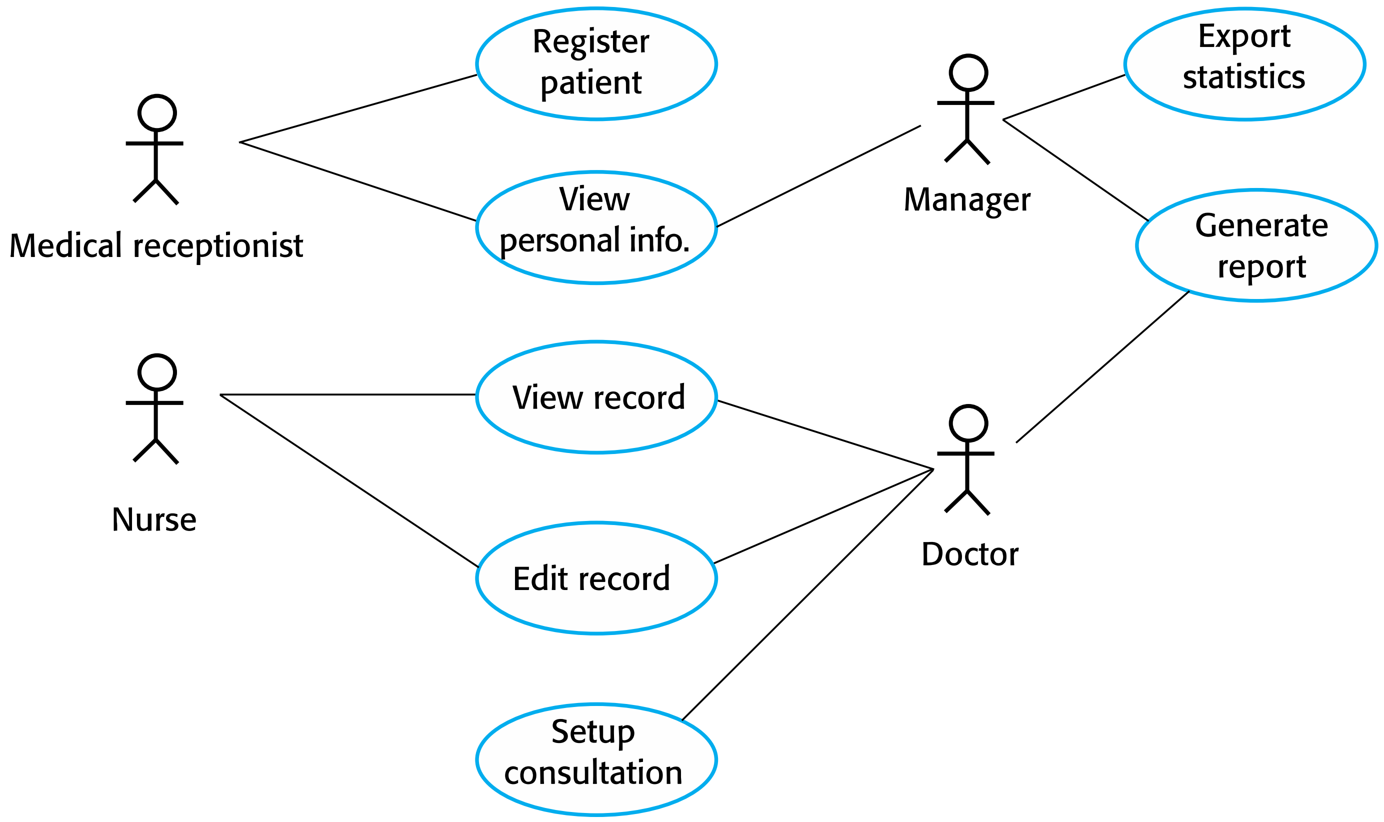
Discovering requirements

- Interaction with system stakeholders and possible end-users
  - Interviewing
    - Closed interviews based on pre-determined list of questions
    - Open interviews : various issues discussed
  - Scenarios
    - Real life examples how a system can be used
    - Should include
      - A description of the starting situation;
      - A description of the normal flow of events;
      - A description of what can go wrong;
      - Information about other concurrent activities;
      - A description of the state when the scenario finishes.

Use cases

- A scenario based technique in the UML which identify the actors in an interaction and which describe the interaction itself
- A set of use cases should describe all possible interactions with the system
- Sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system

Use cases for the MHC-PMS



Requirements validation

- Concerned with demonstrating that the requirements define the system that the customer really wants.
- Requirements error costs are high so validation is very important
  - Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error.

Requirements checking

- Validity. Does the system provide the functions which best support the customer’s needs?
- Consistency. Are there any requirements conflicts?
- Completeness. Are all functions required by the customer included?
- Realism. Can the requirements be implemented given available budget and technology
- Verifiability. Can the requirements be checked?

Requirements validation techniques

- Requirements reviews
  - Systematic manual analysis of the requirements.
- Prototyping
  - Using an executable model of the system to check requirements.
- Test-case generation
  - Developing tests for requirements to check testability.

Requirement reviews

- Regular reviews should be held while the requirements definition is being formulated.
- Both client and contractor staff should be involved in reviews.
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage.

Exercise

- Use Software Specification template to define an application or any other software.

References

- Software Engineering, 9th Edition by Ian Sommerville