SOFTWARE ENGINEERING

Chapter 4 – Requirements Engineering

WEEK 3, 4

Topics covered

- Functional and non-functional requirements
- The software requirements document
- Requirements specification
- Requirements engineering processes
- Requirements elicitation and analysis
- Requirements validation
- Requirements management

REQUIREMENTS

Requirements engineering

- The process of establishing the ***services*** that the customer requires from a system and the ***constraints*** under which it operates and is developed.

What is a requirement?

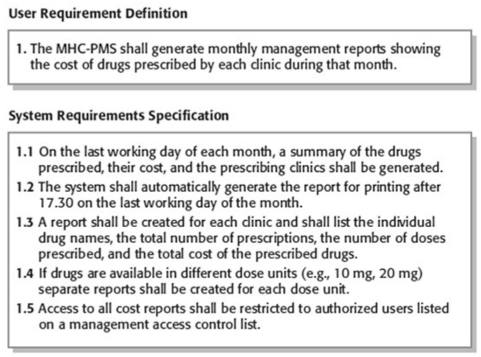
Requirement engineering = process of establishing the ***services*** that the customer requires from a system and the ***constraints*** under which it operates and is developed.

- Requirement = the descriptions of
  - the system services
  - and constraints
- It may range
  - from a high-level abstract statement
  - to a detailed mathematical functional specification.
- May serve a dual function
  - The basis for a bid for a contract - must be open to interpretation;
  - The basis for the contract itself - must be in detail;

Types of requirement

- User requirements
  - Written for customers.
    - statements in natural language + diagrams of the services the system provides and its operational constraints.
- System requirements
  - (For) developers/contractor
    - detailed descriptions of the system’s functions, services and operational constraints

User and system requirements example



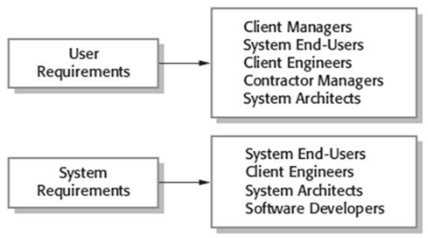
User Requirement Definition

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., 10 mg, 20 mg) 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.

Readers of different types of requirements specification



Functional and non-functional requirements

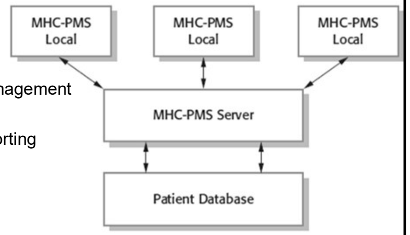
- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
  - May state what the system should not do.
- Non-functional requirements
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
  - Often apply to the system as a whole rather than individual features or services.

Functional requirements

- Describe functionality or system services.
  - Functional user requirements may be high-level statements of what the system should do.
  - Functional system requirements should describe the system services in detail.

Case study: MHC-PMS

- The MHC-PMS (Mental Health Care-Patient Management System) is an information system for clinics.



- Key features:
  - Individual care management
  - Patient monitoring
  - Administrative reporting
- Concerns:
  - Privacy
  - Safety

Functional requirements for the MHC-PMS

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

Requirements imprecision

- Problems arise when requirements are not precisely stated.
  - Ambiguous requirements may be interpreted in different ways by developers and users.
- For the term ‘search’ in requirement 1
  - User intention – search for a patient name across all appointments in all clinics;
  - Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

Requirements completeness and consistency

- Requirements should be both complete and consistent.
- Complete
  - They should include descriptions of all facilities required.
- Consistent
  - There should be no conflicts or contradictions in the descriptions of the system facilities.

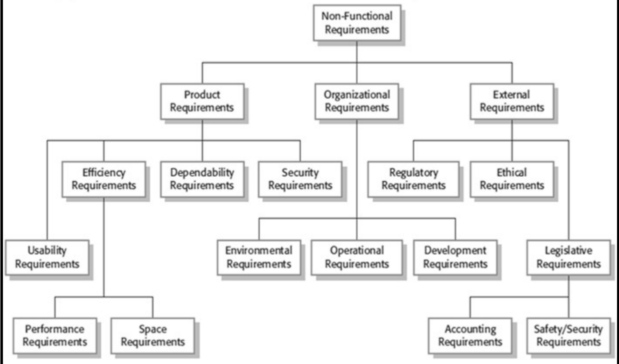
Non-functional requirements

- Define system properties and constraints
  - Properties: reliability, response time and storage requirements.
  - Constraints: I/O device capability, system representations, etc.
- Non-functional requirements may be more critical than functional requirements.
  - If these are not met, the system may be useless.

Non-functional requirements implementation

- Non-functional requirements may affect the overall architecture of a system
  - rather than the individual components.
- A single non-functional requirement
  - may generate a number of related functional requirements
  - and may also generate requirements that restrict existing requirements.

Types of nonfunctional requirement



Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Examples of nonfunctional requirements in 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.

Goals vs. requirements

- Goal
  - A general intention of the user such as ease of use.
- Testable non-functional requirement
  - A statement using some measure that can be objectively tested.

Goals vs. requirements (cont.)

Goal:
The system should be <u>easy to use</u> by medical staff.

Non-functional requirement:
Medical staff shall be <u>able to use</u> all the system functions <u>after four hours of training</u>.

Domain requirements

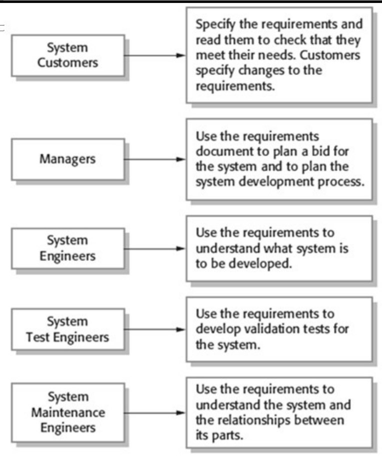
- The system's operational domain imposes requirements on the system.
  - For example, a train control system has to take into account the braking characteristics in different weather conditions.
- Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.
- If domain requirements are not satisfied, the system may be unworkable.

REQUIREMENTS SPECIFICATION

The software requirements document

- The software requirements document is the official statement of what is required of the system developers.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.

Users of a requirements 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.

The structure of a requirements document

Chapter	Description
Preface	define the expected readership of the document and describe its version history
Introduction	describe the need for the system, briefly describe the system's functions, how the system fits into the overall business or strategic objectives
Glossary	define the technical terms used in the document
User requirements definition	describe the services provided for the user, the nonfunctional system requirements; may use natural language, diagrams, other notations that are understandable to customers; product and process standards that must be followed
System architecture	present a high-level overview of the anticipated system architecture, the distribution of functions across system modules

The structure of a requirements document

Chapter	Description
System requirements specification	describe the functional and nonfunctional requirements in more detail
System models	might include graphical system models showing the relationships between the system components and the system and its environment
System evolution	describe the fundamental assumptions, any anticipated changes due to hardware evolution, changing user needs, …
Appendices	provide detailed, specific information that is related to the application being developed
Index	May include several indexes to the document, a normal alphabetic index; may be an index of diagrams, an index of functions,…

Requirements specification

- The process of writing down the user and system requirements in a requirements document.
- Notes:
  - User requirements have to be understandable by end-users and customers who do not have a technical background.
  - System requirements are more detailed requirements and may include more technical information.
  - The requirements may be part of a contract for the system development

Ways of writing a system requirements specification

Notation	Description
Natural language	Sentences in natural language. Each sentence should express one requirement.
Structured natural language	Natural language statements on a standard form or template
Design description languages	Uses a language like a programming language, but with more abstract features
Graphical notations	Graphical models, supplemented by text annotations (best for functional requirements); UML use case and sequence diagrams are commonly used.
Mathematical specifications	Based on mathematical concepts such as finite-state machines or sets; Can reduce the ambiguity but hard to understand (and hard to check manually)

Natural language specification

- Used for writing requirements because it is expressive, intuitive and universal.
  - he requirements can be understood by users and customers.
- Problems
  - Lack of clarity: Precision is difficult without making the document difficult to read.
  - Requirements confusion: Functional and non-functional requirements tend to be mixed-up.
  - Requirements amalgamation: Several different requirements may be expressed together.

Example requirements for the insulin pump software system

- Req 3.2. The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes.
- Req 3.6. The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1.

Structured specifications

- Writing on a standard form or template:
  - Name
  - Inputs, outputs
  - The information needed for the computation
  - Action
  - Pre and post conditions (if appropriate)
  - The side effects (if any)
- This works well for some types of requirements e.g. requirements for embedded control system

A structured specification of a requirement for an insulin pump

Insulin Pump/Control Software/SRS/3.3.2	
Function	Compute insulin dose: safe sugar level
Description	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.
Inputs	Current sugar reading (r2); the previous two readings (r0 and r1).
Source	Current sugar reading from sensor. Other readings from memory.
Outputs	CompDose—the dose in insulin to be delivered
Destination	Main control loop.
Action	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.
Requirements	Two previous readings so that the rate of change of sugar level can be computed
Pre-condition	The insulin reservoir contains at least the maximum allowed single dose of insulin.
Post-condition	r0 is replaced by r1 then r1 is replaced by r2.
Side effects	None

Tabular specification

- Particularly useful when you have to define a number of possible alternative courses of action.
- Example:

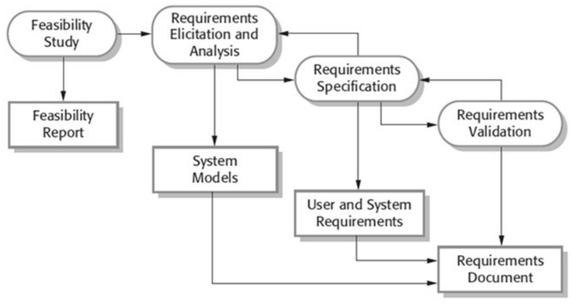
Condition	Action
Sugar level falling (r2 < r1)	CompDose = 0
Sugar level stable (r2 = r1)	CompDose = 0
Sugar level increasing and rate of increase decreasing ((r2 – r1) < (r1 – r0))	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ((r2 – r1) ≥ (r1 – r0))	CompDose = round ((r2 – r1)/4) **If** rounded result = 0 **then** CompDose = MinimumDose

REQUIREMENTS ENGINEERING PROCESSES

Requirements engineering processes

- Processes to “generate” all requirements
- Generic activities common to all processes
  - Requirements elicitation;
  - Requirements analysis;
  - Requirements validation;
  - Requirements management.
- In practice, RE is an iterative activity

Do you remember this?



A spiral view of the requirements engineering process

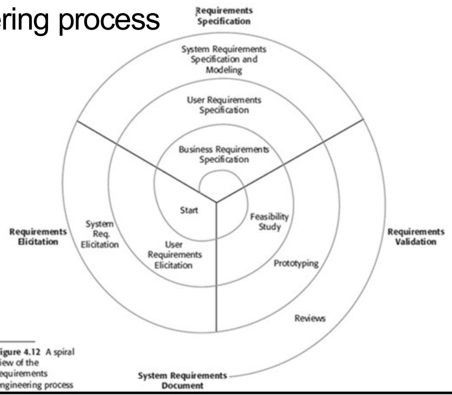
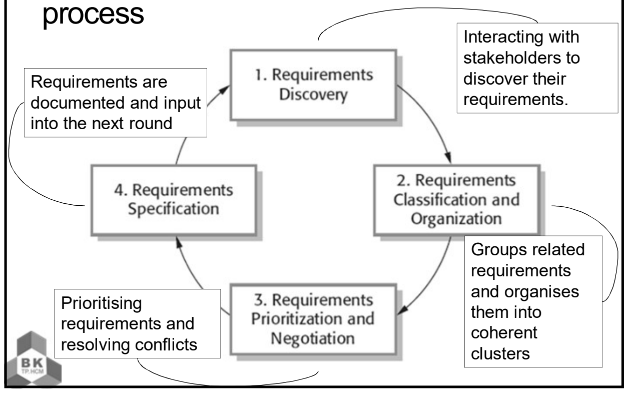

Requirements Specification
System Requirements Specification and Modeling
User Requirements Specification
Business Requirements Specification
Start
Feasibility Study
Requirements Elicitation
System Req. Elicitation
User Requirements Elicitation
Prototyping
Requirements Validation
Reviews
System Requirements Document

Figure 4.12 A spiral view of the requirements engineering process

Requirements elicitation and analysis

- ~ requirements elicitation or requirements discovery.
- Work with customers to find out:
  - the application domain, the services and the operational constraints (system performance, hardware constraints, etc.).
- May involve
  - end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called ***stakeholders***.

The requirements elicitation and analysis process


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritization and Negotiation
4. Requirements Specification

Interacting with stakeholders to discover their requirements.

Groups related requirements and organises them into coherent clusters

Prioritising requirements and resolving conflicts

Requirements are documented and input into the next round

Problems of requirements elicitation

- Stakeholders don't know what they really want.
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organisational and political factors may influence the system requirements.
- The requirements change during the analysis process.
  - New stakeholders may emerge and the business environment change.

Requirements discovery

- To gather information about the required and existing systems and distil the user and system requirements from this information.
- Main concerns:
  - Stakeholders
  - Discovery techniques/approaches/…

Stakeholders in the MHC-PMS

Stakeholder	Why? - Role
Patients	whose information is recorded in the system
Doctors	responsible for assessing and treating patients
Nurses	coordinate the consultations with doctors and administer some treatments
Medical receptionists	manage patients' appointments
IT staff	responsible for installing and maintaining the system
Medical ethics manager	ensure that the system meets current ethical guidelines for patient care
Health care managers	obtain management information from the system
Medical records staff	responsible for ensuring that system information can be maintained and preserved, and that record keeping procedures have been properly implemented.

Discovery technique - Interviewing

- Part of most RE processes.
- Types of interview
  - Closed vs Open => mixed?
- Be effective
  - Be open-minded, avoid pre-conceived ideas about the requirements and are willing to listen to stakeholders.
  - Prompt the interviewee to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system.

Scenarios

- Scenarios are real-life examples of how a system can be used.
- They 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.

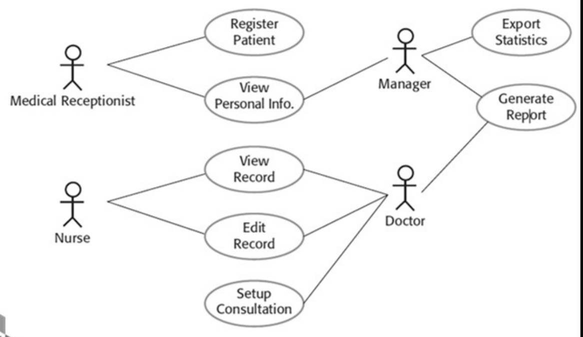
Scenario for collecting medical history in MHC-PMS

- **Initial assumption:**
  - The patient has seen a medical receptionist who has created a record in the system and collected the patient's personal information (name, address, age, etc.). A nurse is logged on to the system and is collecting medical history.
- **Normal:**
  - The nurse searches for the patient by family name. If there is more than one patient with the same surname, the given name (first name in English) and date of birth are used to identify the patient.
  - The nurse chooses the menu option to add medical history.
  - The nurse then follows a series of prompts from the system to enter information about consultations elsewhere on mental health problems (free text input), existing medical conditions (nurse selects conditions from menu), medication currently taken (selected from menu), allergies (free text), and home life (form).
- **What can go wrong:**
  - The patient's record does not exist or cannot be found. The nurse should create a new record and record personal information.
  - …
- …

Use cases

- Use-cases are 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.

Use cases for the MHC-PMS



Ethnography

- Observational technique
  - used to understand operational processes and help derive support requirements for these processes
- How
  - A social scientist spends a considerable time observing and analysing how people actually work.
  - People do not have to explain or articulate their work.
  - Social and organisational factors of importance may be observed.

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.

Some guidelines for writing a Detailed Requirement

- 1. Classify requirement as functional or non-functional
- 2. Size carefully
- 3. Make trace-able if possible
- 4. Make testable
- 5. Make sure not ambiguous
- 6. Give the requirement a priority
- 7. Check that requirement set complete
- 8. Include error conditions
- 9. Check for consistency

Requirements management

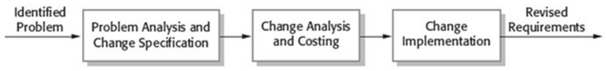
- The process of managing changing requirements during the requirements engineering process and system development.
- Why changes?
  - The business and technical environment of the system always changes after installation.
  - The people who pay for a system and the users of that system are rarely the same people.
  - Large systems usually have a diverse user community, with many users having different requirements and priorities that may be conflicting or contradictory.

Requirements management planning

- Establishes the level of requirements management detail that is required.
- Requirements management decisions:
  - Requirements identification
  - A change management process
  - Traceability policies
  - Tool support

Requirements change management

- Deciding if a requirements change should be accepted
  - Problem analysis and change specification
  - Change analysis and costing
  - Change implementation


Identified Problem
Problem Analysis and Change Specification
Change Analysis and Costing
Change Implementation
Revised Requirements

Summary

- Requirements: what the system should do and constraints on its operation and implementation.
- Functional requirements = the services
- Non-functional requirements = constraints (development & use)
  - apply to the system as a whole.
- The software requirements document (i.e. SRS) is an agreed statement of the system requirements.
- The RE process is an iterative process
  - requirements elicitation, specification and validation.

Summary (cont.)

- Requirements elicitation and analysis = iterative process
  - requirements discovery, classification and organization, negotiation and requirements documentation.
- Techniques for requirements elicitation
  - interviews, scenarios, use-cases and ethnography, etc.
- Requirements validation = checking the requirements
  - for validity, consistency, completeness, realism and verifiability.
- Business, organizational and technical changes inevitably
  - => changes to the requirements for a software system.
- Requirements management = managing and controlling the requirement changes.