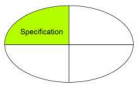
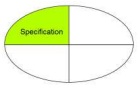


Chapter 4 – Requirements Engineering

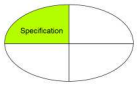
Successful Software Development, Donaldson and Siegel, 2001

Topics covered



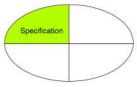
- ✧ Intro
- ✧ 4.1 Functional and non-functional requirement
- ✧ 4.2 The software requirements document

Requirements engineering



- ✧ Requirements:
  - Description what the system should or should not do (functionality, constraints)
- ✧ Requirements Engineering:
  - Process of finding out, analyzing, documenting, and checking requirements

Requirement Documents



high-level, abstract statement . . . . . detailed specification

Dual function of requirements document:

- basis for a bid for a contract
  => open to interpretation;
- basis for implementation of software
  => defined in detail;

2 types of requirement

- ✧ User requirements
  - natural language + diagrams
  - Less detailed
  - e.g. For managers
- ✧ System requirements
  - Structured, detailed document
  - May be part of a contract between client and software developer.
  - e.g. For developers

User and system requirements

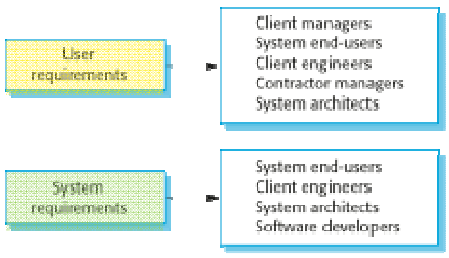
User requirement definition

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

System requirements specification

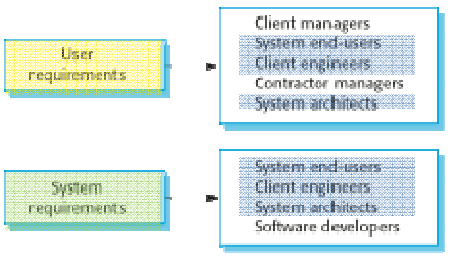
1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall automatically generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic 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.

Readers of different types of requirements specification



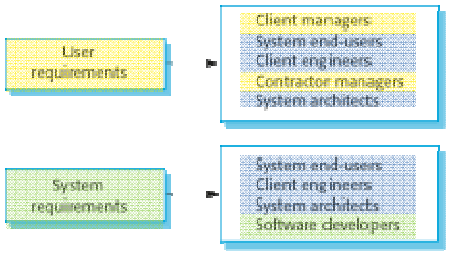
7

Readers of different types of requirements specification



8

Readers of different types of requirements specification



9

4.1 Functional vs Non-Functional Requirements

Functional Requirements

Non-functional requirements

- Domain requirements

10

4.1.1 Functional requirements

- Specific facilities provided by system
- Should be complete and consistent

11

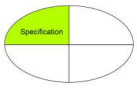
Examples:

✧ 3 Functional requirements for the MHC-PMS

- Users shall be able to search the appointments lists for all clinics.
- 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.

13

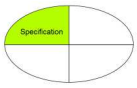
Examples:



✧ 3 Functional requirements for the MHC-PMS

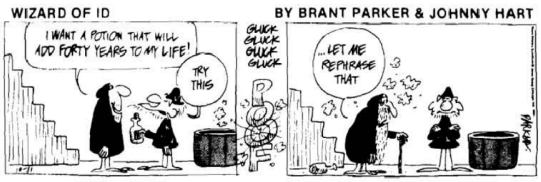
- Users shall be able to search the appointments lists for all clinics. Ambiguity!
- System shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
- Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

Examples:



✧ 3 Functional requirements for the MHC-PMS

- Users shall be able to search the appointments lists for all clinics. Ambiguity!
- 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. Ambiguity!



Successful Software Development, Donaldson and Siegel, 2001

That happens . . .

I know you believe you understood what you think I said, but I am not sure you realize that what you heard is not what I meant."

Completeness and Consistency

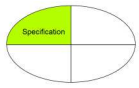
✧ Principle vs Practice

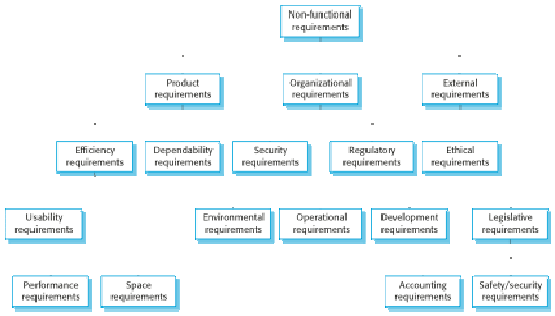
- In principle, requirements should be both **complete** and **consistent**.
- In practice, it is impossible to produce a 100% complete and consistent requirements document.

4.1.2 Non-Functional requirements

- Not directly related to specific facilities
- Usually affect system as a whole
  - Emergent system properties (e.g. Reliability, Performance)
  - Constraints on system (e.g. Regulations)

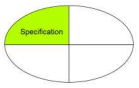
Types of nonfunctional requirement

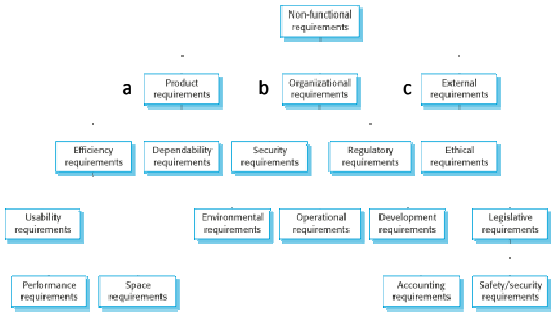




20

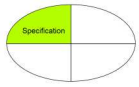
Types of nonfunctional requirement





21

Examples of nonfunctional requirements :



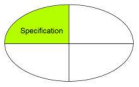
a **Product requirement**
The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 08:30–17:30). Downtime within normal working hours shall not exceed five seconds in any one day.

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

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

22

Non-Functional requirements



- May affect system architecture (e.g. Performance)
- A single non-functional requirement may generate multiple functional requirements (e.g. Security)
- Often more critical than functional requirements

23

Common problem with non-functional requirements:

✧ Goals (general intention) instead of verifiable non-f. requirements

24

Common problem with non-functional requirements:

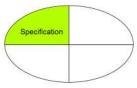
✧ Goals (general intention) instead of verifiable non-f. requirements

✧ Goal example:

- The system should be easy to use by medical staff and should be organized in such a way the user errors are minimized

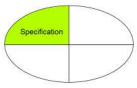
25

Common problem with non-functional requirements:



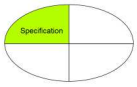
- Goals (general intention) instead of verifiable non-f. requirements
- Goal example:
  - The system should be easy to use by medical staff and should be organized in such a way the user errors are minimized
- Verifiable non-functional requirement example:
  - Medical staff shall be able to use all the system functions after four hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use

Common problem with non-functional requirements:



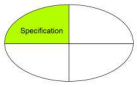
- Goals (general intention) instead of verifiable non-f. requirements
- Goal example:
  - The system should be easy to use by medical staff and should be organized in such a way tha user errors are minimized
- Verifiable non-functional requirement example:
  - Medical staff shall be able to use all the system functions after four hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use quantitative!

Metrics for specifying nonfunctional requirements (Fig 4.5)



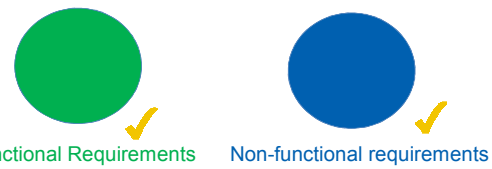
Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Difficulties can arise ..



- Hard to quantify
- Quantification not meaningful to stake-holder
- Expensive to measure

4.1 Functional vs Non-Functional Requirements



Functional Requirements

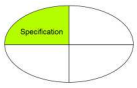
Non-functional requirements

- Domain requirements

Domain requirements

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

Domain requirement for a train protection system

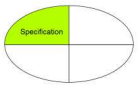


✧ The deceleration of the train shall be computed as:

- **Dtrain = Dcontrol + Dgradient**
- where Dgradient is 9.81ms2 * compensated gradient/alpha and where the values of 9.81ms2 /alpha are known for different types of train.

32

Problems with domain requirements



✧ Understandability

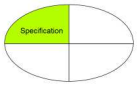
- Requirements expressed in language of the application domain;

✧ Implicitness

- Domain specialists understand the area so well that they might not think of making the domain requirements explicit.

33

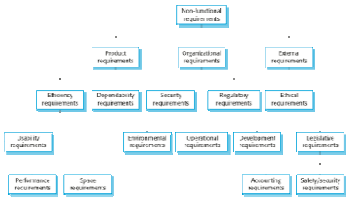
TODO:



Brain storm

- functional
- non-functional

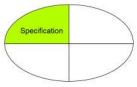
requirements

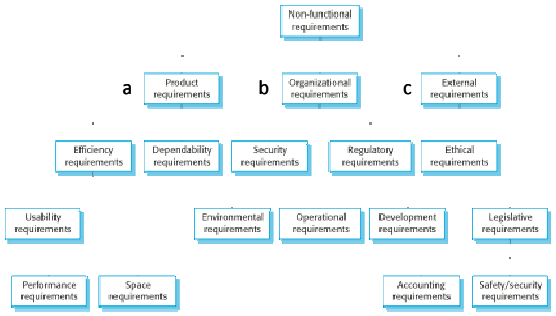


for a new 'virtual campus' (Canvas / BlackBoard)

34

Types of nonfunctional requirement





35