FRESNO STATE
Discovery. Diversity. Distinction.

CSCI 150
Intro to Software Engineering

October 18, 2018

Shih-Hsi “Alex” Liu

Chapter 4 – Requirements Engineering

Lecture 1

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 engineering

- ✧ Requirements are the descriptions of the system *services and constraints*. They reflect the needs of customers for a system that serves a certain purpose.

- ✧ 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?

- ✧ It may range from a high-level *abstract* statement of a service or of a system constraint to a *detailed* mathematical functional <u>specification (some textbooks separate specification from requirements)</u>.
- ✧ This is inevitable as requirements may serve a dual function
  - May be the basis for a bid for a contract - therefore must be <u>open to interpretation;</u>
  - May be the basis for the contract itself - therefore must be <u>defined in detail (called specification in some textbooks)</u>;
  - Both these statements may be called requirements.

Requirements abstraction (Davis)

"If a company wishes to let a contract for a large software development project, it must define its needs in a sufficiently abstract way that a solution is not pre-defined. The requirements must be written so that several contractors can bid for the contract, offering, perhaps, different ways of meeting the client organization's needs. Once a contract has been awarded, the contractor must write a system definition for the client in more detail so that the client understands and can validate what the software will do. Both of these documents may be called the requirements document for the system."

Types of requirement

- ✧ User requirements (called requirements documents in some textbooks)
  - Statements in <u>natural language plus diagrams</u> of the services the system provides and its operational constraints. *Written for customers.*
- ✧ System requirements (called functional specification in some texts)
  - A <u>structured</u> document setting out <u>detailed descriptions</u> of the system’s functions, services and operational constraints. Defines what should be implemented so may be part of a <u>contract</u> between client and contractor.

Figure 4.1 User and system requirements (How close are they to your step-by-step descriptions?)

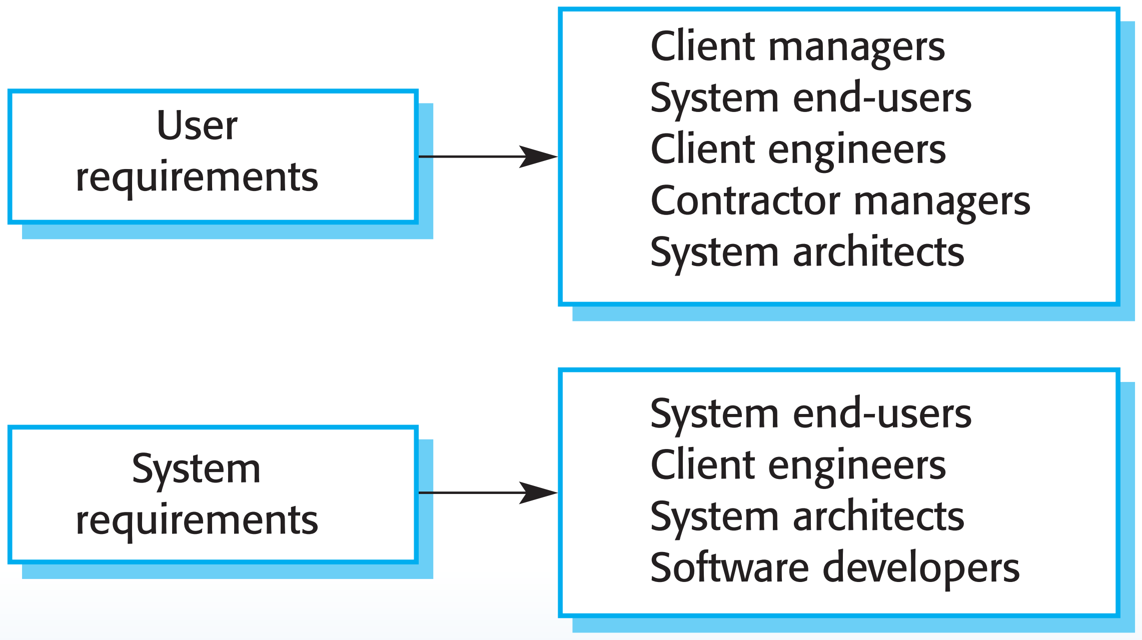
User requirement definition

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.

Figure 4.2 Readers of different types of requirements specification



Types of requirement

- The readers of user requirements are not usually concerned with how the system will be implemented.
- The readers of system requirements need to know more precisely what the system will do because they are concerned with how it will support the business processes or because they are involved in the system implementation.

4.1 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 and non-functional requirements

- Functional Req. and non-functional Req. cannot really separate clearly
  - One functional req. may have effect on other non-functional req. and vice versa (e.g., security).
- System req. are not independent and that one req. often generates or constrains other req. (called tangling req.) Therefore, system req. do not just specify services or features of the system; they also specify the necessary functionality to ensure these services/features are delivered properly.
- Domain requirements
  - Constraints on the system from the domain of operation

Functional requirements

- ✧ Describe functionality or system services.
  - ✧ Namely, functional req. describe what system should do in different levels of details.
- ✧ Depend on the type of software, expected users and the type of system where the software is used.
- ✧ Functional user requirements may be high-level statements of what the system should do.
- ✧ Functional system requirements should describe the system services in detail.

Functional 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.
- What do you observe from the above 3 requirements?
  - How do you interpret the first requirement?

Requirements imprecision

- ✧ Problems arise when requirements are not precisely stated.
- ✧ Ambiguous requirements may be interpreted in different ways by developers and users.
- ✧ Consider the term ‘search’ in requirement 1
  - ▪ User intention – search for a patient name **across all appointments** in **all clinics**; (because patients are mentally ill and may get confused).
  - ▪ Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.
- ▪ Developers tend to simplify the interpretation so that the implementation will be easier. But in most cases, it is not what customer wants.

Requirements completeness and consistency

- In principle, 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.
- In practice, it is impossible to produce a complete and consistent requirements document.

Case study: Elevator Problem

- See next slide
  - An *n* elevator system to be installed in a building with *m* floors
  - Natural language specifications contain several <u>ambiguities</u> and <u>inconsistencies</u>

Case study: Elevator Problem

A product is to be installed to control *n* elevators in a building with *m* floors. The problem concerns the logic required to move elevators between floors according to the following constraints:

1. Each elevator has a set of *m* buttons, one for each floor. These illuminate when pressed and cause the elevator to visit the corresponding floor. The illumination is canceled when the corresponding floor is *visited* by the elevator
2. Each floor has two buttons, one to request an up-elevator, one to request a down-elevator. These buttons illuminate when pressed. The illumination is canceled when an elevator *visits* the floor, then moves in the desired direction
3. If an elevator has no requests, it remains at its current floor with its doors closed
4. The algorithm to decide which to service first should *minimize* the waiting time for both requests

Easy question (?)

- How many buttons are on the first and top floors?

From informal specs…

"The illumination is cancelled when the elevator visits the floo,r then moves in the desired direction"

2 different interpretations (for the case of "up" call)

- switch off as the elevator arrives at the floor from below (obvious restrictions for 1st and last floor)
- switch off after the elevator starts moving up after visiting
  - in practice you may observe the two cases!

…more analysis of informal specs

"The algorithm to decide which to service first should *minimize* the waiting time for both requests."

what does this mean?

- in no other way can you satisfy either request in a shorter time
  - but minimizing for one may require longer for the other
- the sum of both is minimal
  - why the sum?

Non-functional requirements

- ✧ These define system properties and constraints e.g., reliability, response time and storage requirements, I/O device capability, data representations, etc.
- ✧ Process requirements may also be specified mandating a particular IDE, programming language or development method.
- ✧ Overall, req. that is not functional are considered non-functional req.
- ✧ Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be useless.
  - ✧ MHC-PMS, real-time aircraft (reliability), embedded systems (performance), patriot missiles (real-time)

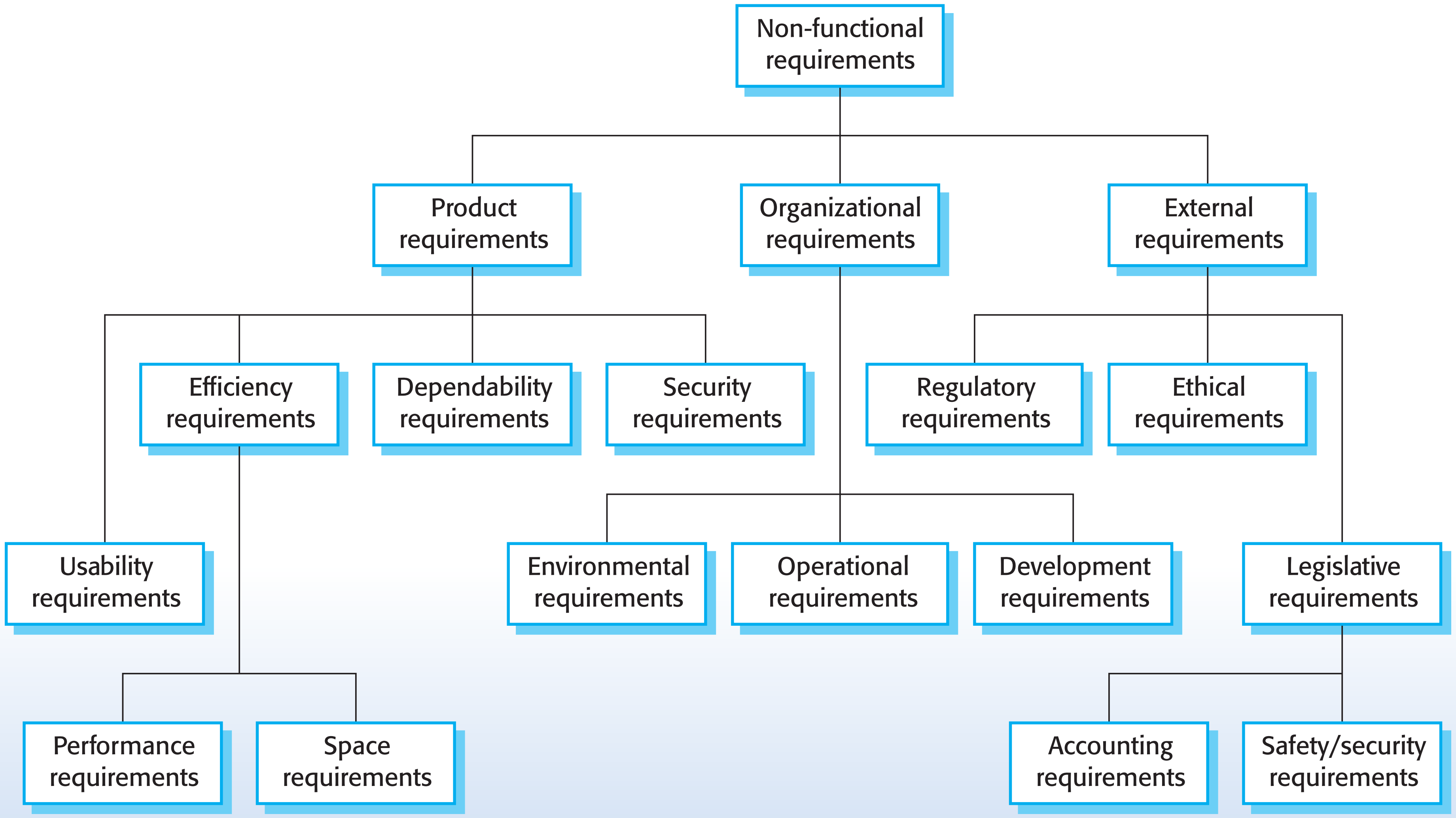
Non-functional Requirements (NFR) implementation

- **It is more difficult to relate components to NFR**.
- The implementation of NFR may be diffused throughout the system (e.g., **crosscutting** concerns introduced in CSCI 252)
- Non-functional requirements may affect the overall **architecture** of a system rather than the individual components.
  - For example, to ensure that performance requirements are met, you may have to organize the system to minimize communications between components.
- A single non-functional requirement, such as a security requirement, may generate a number of related functional requirements that define system services that are required (e.g., authentication and authorization).
  - It may also generate requirements that restrict existing requirements.

Where do NFR come from?

- NFR arise through user needs
  - Budget constraints
  - Organizational policies
  - Need for interoperability with other SW
  - External factors (e.g., safety regulations, privacy legislation (HIPAA)
- Three categories:
  - Product
  - Organizational
  - External

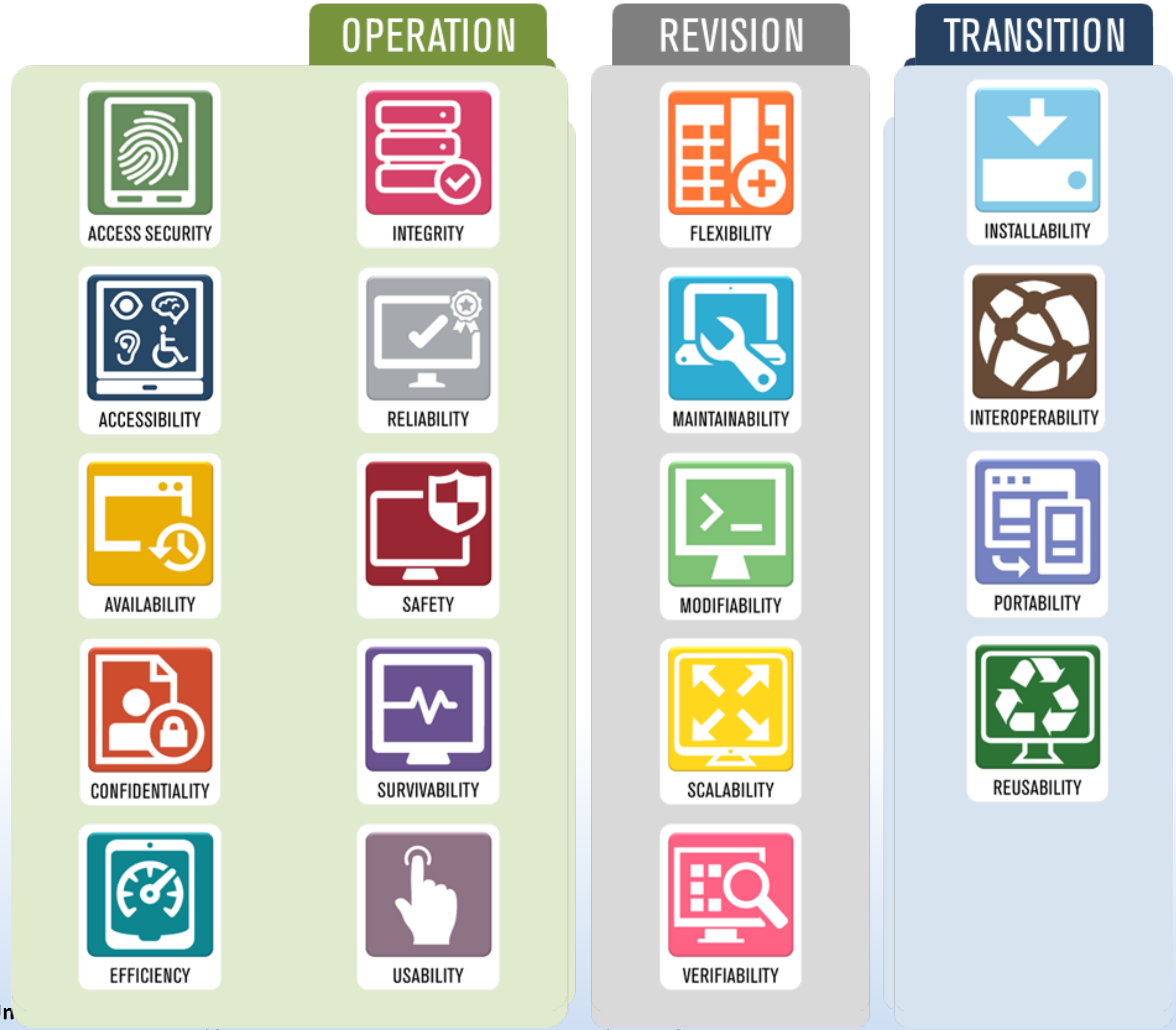
Figure 4.3 Types of nonfunctional requirement



Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way. e.g., execution speed, reliability, memory, security, usability etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures. e.g., process standards used, implementation language requirements, OS 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.

Yet, another classification from industry



https://requirementsquest.com/nonfunctional-requirement-examples/

Yet, another classification from industry

- **OPERATION, or using the functionality.** This user perceives the system as an electronic tool that helps to automate what would otherwise be done manually. From this point of view, the user is concerned with how well the system operates.
- **REVISION, or changing source code or data that drive the system.** The user perceives the system as a set of programmed language statements. These statements are treated as a problem that must be solved. The system must be analyzed, modified, and tested as problems arise, or the business changes the way it operates.
- **TRANSITION, or managing the upkeep of the system.** From this point of view, the system carries similar characteristics as hardware. That is, the user is concerned with aspects such as packaging, transport, and compatibility with other systems.

https://requirementsquest.com/nonfunctional-requirement-examples/

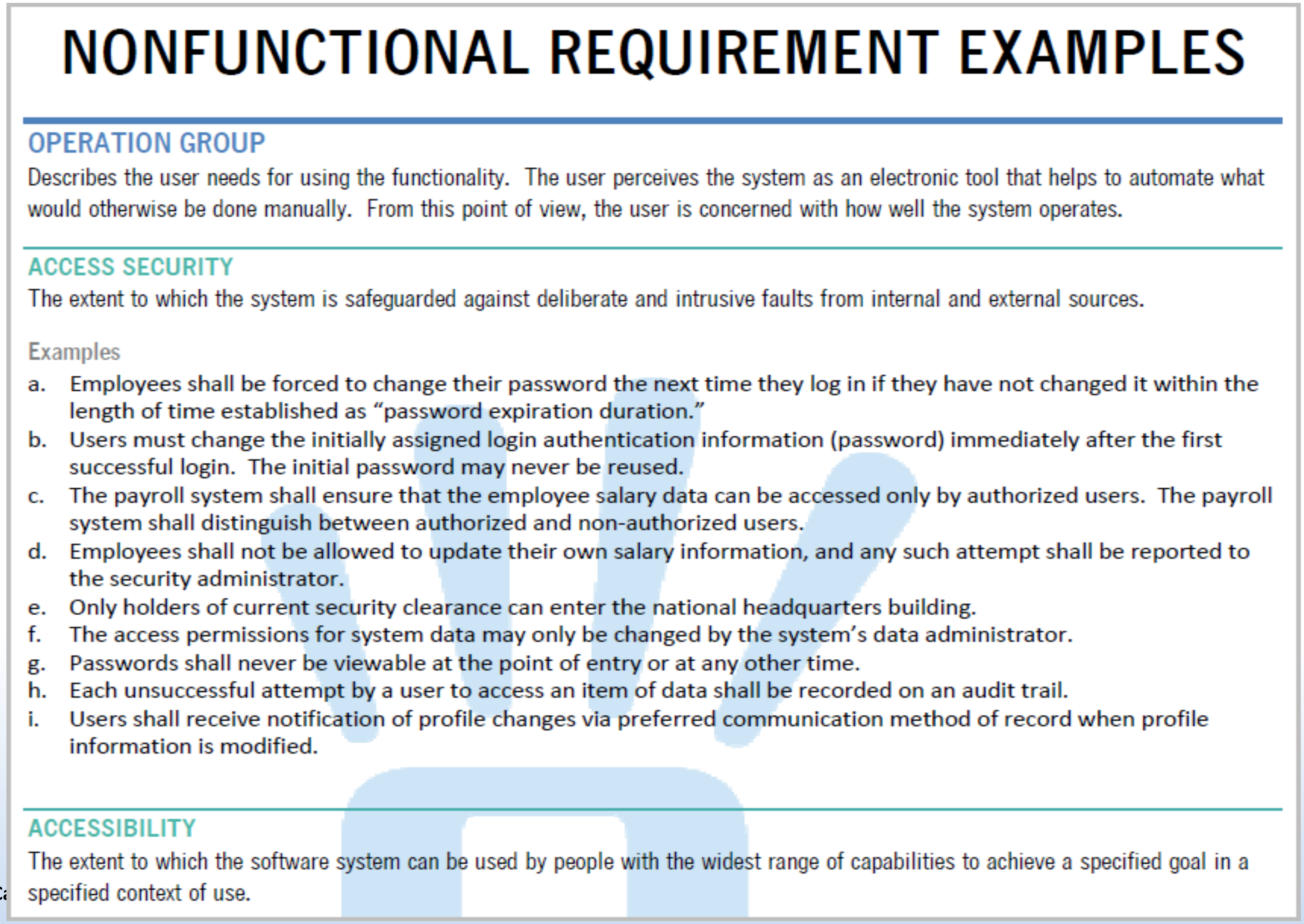
Examples of nonfunctional requirements in the MHC-PMS

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.
(reliability/availability)

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

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

Another example: Security



NONFUNCTIONAL REQUIREMENT EXAMPLES

OPERATION GROUP

Describes the user needs for using the functionality. The user perceives the system as an electronic tool that helps to automate what would otherwise be done manually. From this point of view, the user is concerned with how well the system operates.

ACCESS SECURITY

The extent to which the system is safeguarded against deliberate and intrusive faults from internal and external sources.

Examples

a. Employees shall be forced to change their password the next time they log in if they have not changed it within the length of time established as "password expiration duration."
b. Users must change the initially assigned login authentication information (password) immediately after the first successful login. The initial password may never be reused.
c. The payroll system shall ensure that the employee salary data can be accessed only by authorized users. The payroll system shall distinguish between authorized and non-authorized users.
d. Employees shall not be allowed to update their own salary information, and any such attempt shall be reported to the security administrator.
e. Only holders of current security clearance can enter the national headquarters building.
f. The access permissions for system data may only be changed by the system's data administrator.
g. Passwords shall never be viewable at the point of entry or at any other time.
h. Each unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail.
i. Users shall receive notification of profile changes via preferred communication method of record when profile information is modified.

ACCESSIBILITY

The extent to which the software system can be used by people with the widest range of capabilities to achieve a specified goal in a specified context of use.

Goals and requirements

- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- Goal
  - A general intention of the user such as ease of use. But it may cause problem *because it leaves scope for interpretation and subsequent dispute* once the system is delivered.

Goals and requirements

- ✧ Problematic goal
  - ✧ The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized
- ✧ A better goal
  - ✧ Medical staff should 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.
- ✧ Verifiable non-functional requirement (Quantitative ways)
  - ▪ A statement using some measure that can be objectively tested.
- ✧ Goals are helpful to developers as they convey the intentions of the system users.

Metrics for specifying nonfunctional requirements

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

Importance of security requirements

- https://www.pmi.org/learning/library/importance-of-security-requirements-elicitation-9634

Metrics for specifying nonfunctional requirements

- Some NFR are not quantifiable
  - For example, maintenance, security
  - Even user-defined metrics are introduced (normalized quantities), clients may not relate metrics well with these NFR
  - Also, the cost to measure these NFR is usually high
- NFR often conflict and interact w/ other functional or non-functional req.
  - For example, a reader card is needed if hospital ID cards are used for authentication. But some may wish to have mobile/remote access.
  - Difficult to separate functional and non-functional req.
    - BUT good to highlight emergent system properties

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*.
  - An autopilot system needs to take into account wind direction, temperature, air currents etc.
- Domain requirements be new functional/non-functional requirements, constraints on existing requirements or define specific computations.
- If domain requirements are not satisfied, the system may be unworkable.

Train protection system

- This is a 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.
- It is difficult for a non-specialist to understand the implications of this and how it interacts with other requirements.

Domain requirements problems

- Understandability
  - Requirements are expressed in the language of the application domain;
  - This is often not understood by software engineers developing the system.
- Implicitness
  - Domain specialists understand the area so well that *they do not think of making the domain requirements explicit.*

Key points

- ✧ Requirements for a software system set out what the system should do and define constraints on its operation and implementation.
- ✧ Functional requirements are statements of the services that the system must provide or are descriptions of how some computations must be carried out.
- ✧ Non-functional requirements often constrain the system being developed and the development process being used.
- ✧ They often relate to the emergent properties of the system and therefore apply to the system as a whole.