Software Engineering Theory and Practice

School of Computing	UNIVERSITY OF PORTSMOUTH
Title	Software Engineering Theory and Practice
Module Coordinator	Steven Ossont
Email	steven.ossont@port.ac.uk
Code	M30819
Moodle	https://moodle.port.ac.uk/course/view.php?id=11429

U30819: Software Engineering Theory and Practice

Requirements Engineering

https://moodle.port.ac.uk/course/view.php?id=11429

Steven Ossont https://github.com/orgs/M30819-2020/teams/students/discussion

Based on Ian Somerville, Software Engineering, 10th edition, Chapter 4

Requirements engineering

- The process of establishing
  - the services that the customer requires
  - constraints under which it operates
  - constraints under which it is developed
- The requirements are
  - the descriptions of the system services
  - constraints that result from the requirements process

A requirement ...

A requirement can be used in different places:

- Contract bid (Abstract: For early negotiations, multiple bids)
    - High-level abstract statement
- Contract (Defined in detail)
    - Detailed specification

Requirements

"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."
(Davis)

User vs System Requirements

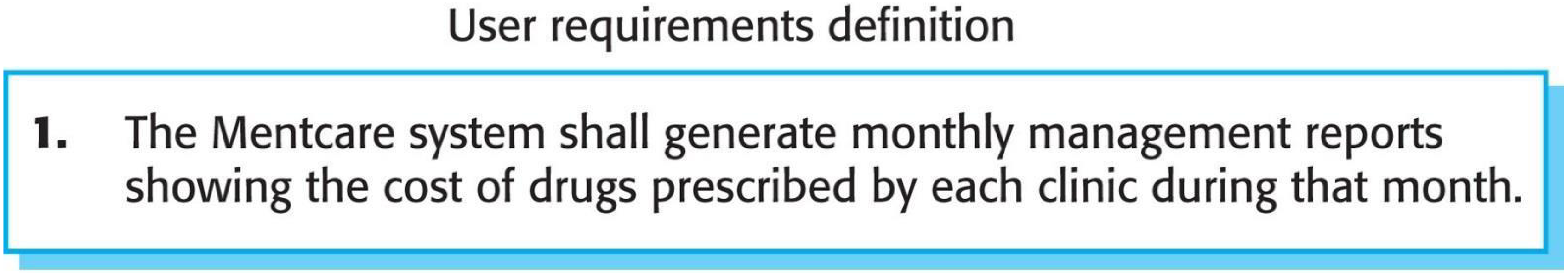
- User requirements
  - Statements in natural language
  - diagrams of the services
  - operational constraints. (Written for customers)

User vs System Requirements

- System requirements
  - structured document
  - detailed descriptions of
    - functions
    - services
    - operational constraints
  - defines what should be implemented

> This can form a contract between client and contractor.

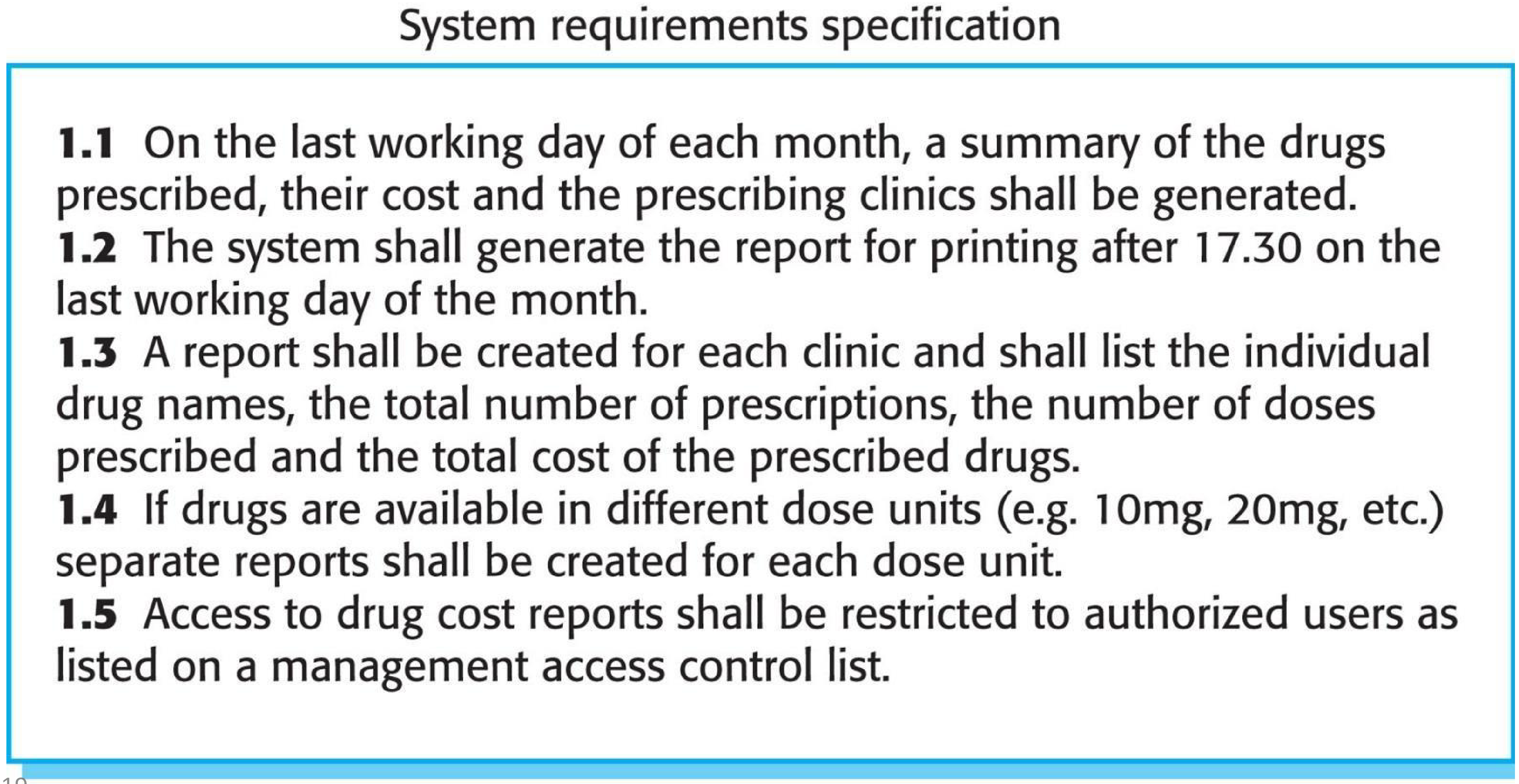
Example User Requirement



User requirements definition

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

Example System Requirement



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 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.

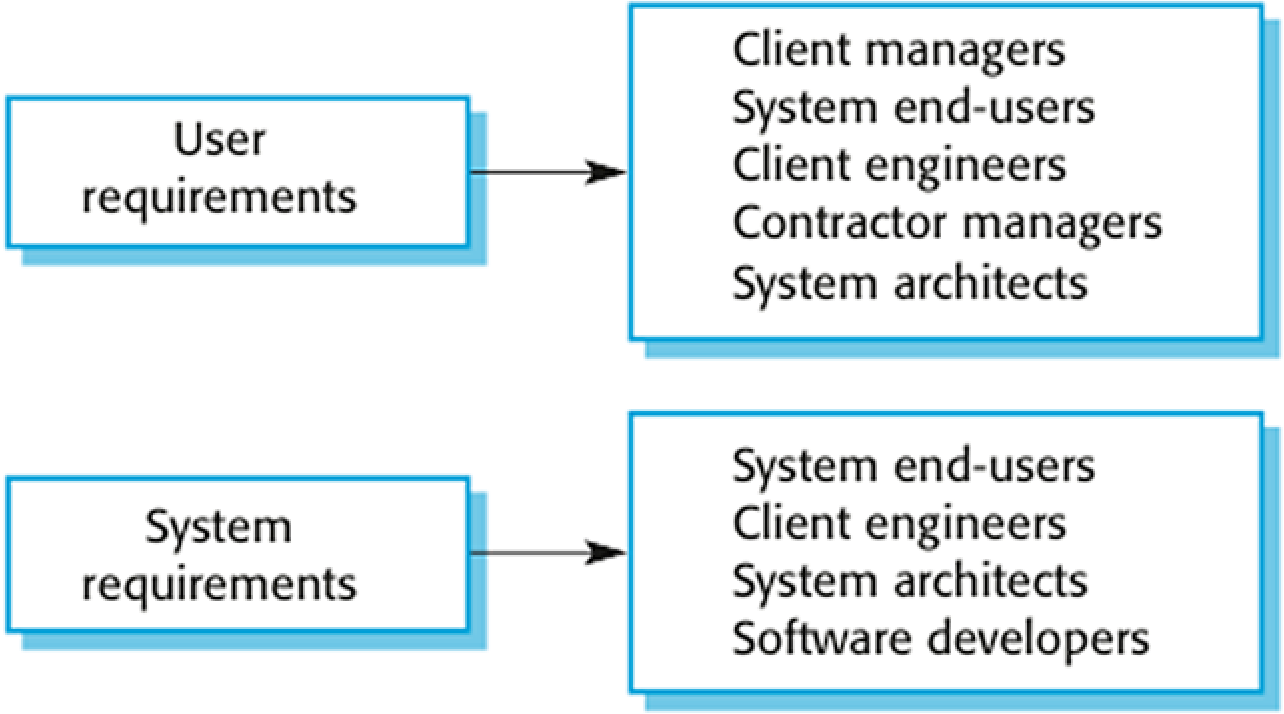
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.

1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

Example System Requirement

- What is the structure of such a report?
- How is the cost represented?
- Who can access these reports?
- What day of the month/time of the day will they be generated?
- How fast should these reports be generated?
- How should drugs be identified?
- How are drugs with the same name, but different dosage represented?

Who is the audience



> Requirement *wish* (User) vs a testable requirement (system)

Requirements imprecision

- Problems arise when requirements are not precisely stated
- Ambiguous requirements (developers and users have a different interpretation)

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.

> For a sizable project it is impossible to produce a complete and consistent requirements document.

Functional vs Non-functional Requirements

Functional

- Statements of services the system should provide
- How the system should react to inputs
- How the system should behave
- What the system should not do

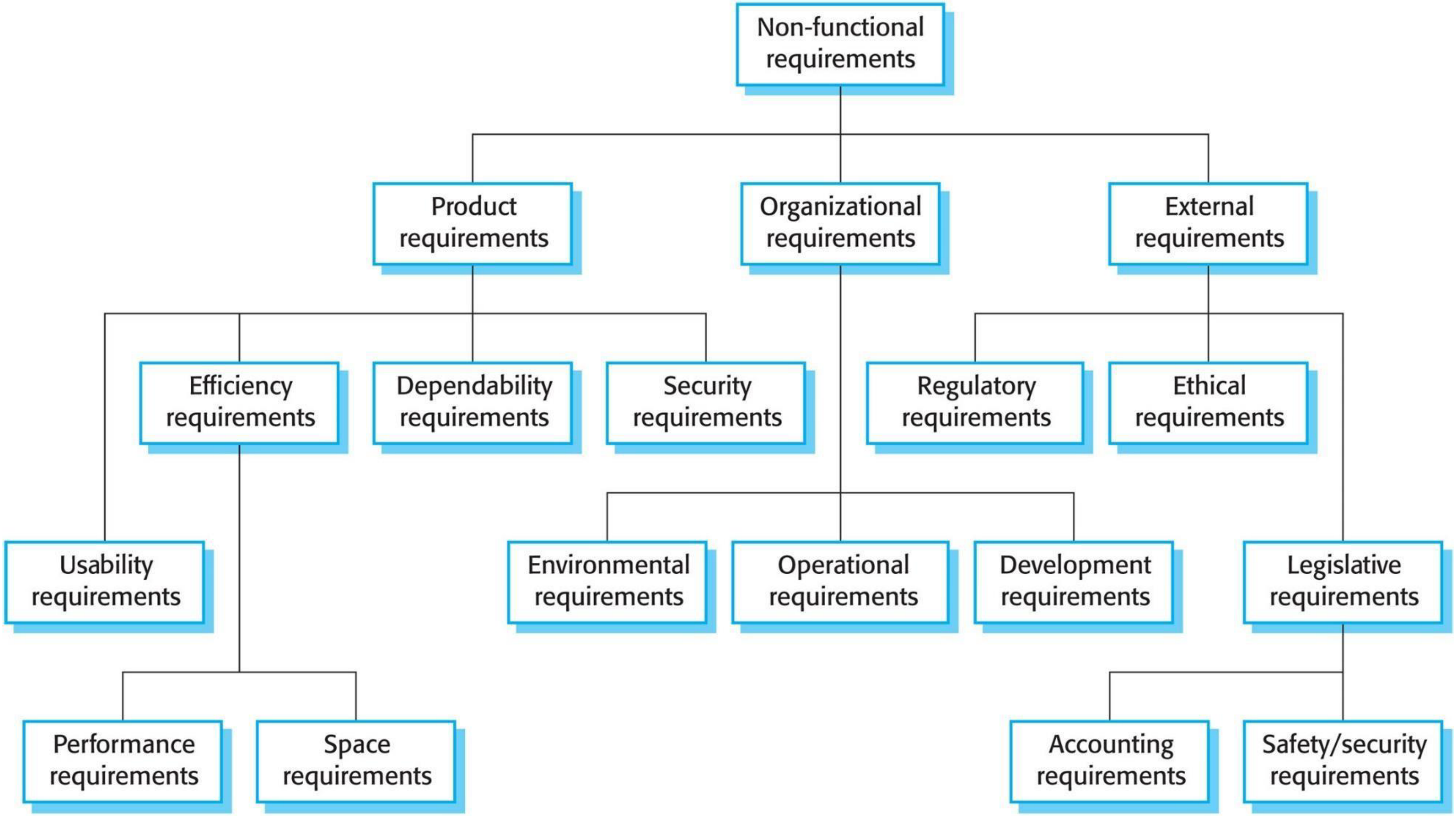
Non-Functional

- Constraints on the system services
- Constraints on the system functions

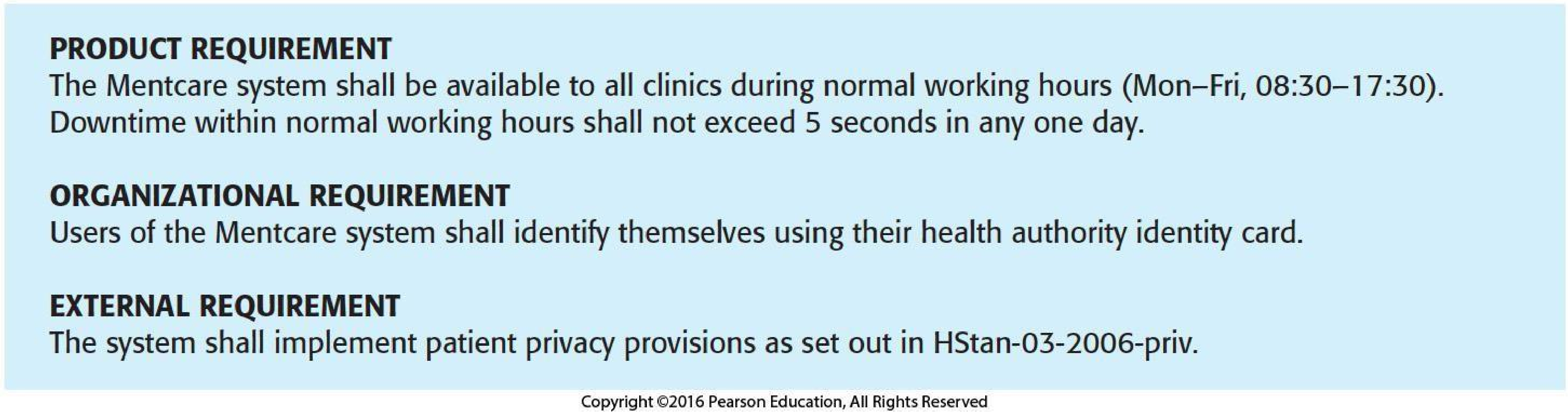
Example Functional Requirements

- 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 eight-digit employee number.

Non-functional Requirements



Example Non-Functional Requirements



PRODUCT REQUIREMENT
The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 08:30–17:30). Downtime within normal working hours shall not exceed 5 seconds in any one day.

ORGANIZATIONAL REQUIREMENT
Users of the Mentcare system shall identify themselves using their health authority identity card.

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

> Are you aware of GDPR?

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
- External requirements
    - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements

Non-Functional Requirements

System properties and constraints:

- reliability
- response time
- storage requirements.

Process requirements:

- particular IDE
- programming language
- development method.

> Non-functional requirements may be more critical than functional requirements. If these are not met, the system may be **useless**.

Usability Requirement

User requirement goal

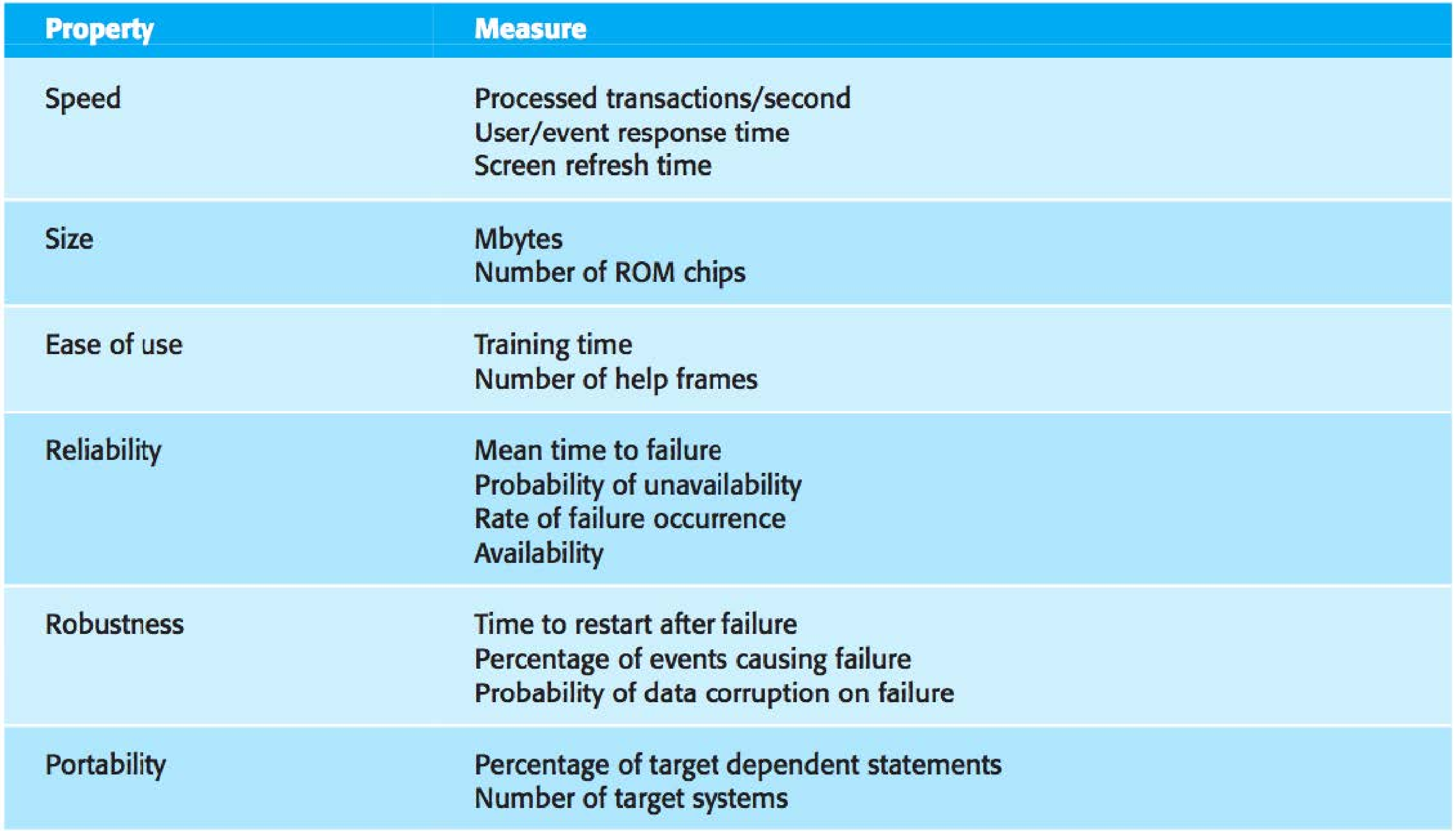
- The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized.

Non functional requirement

- 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.

> 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

Requirements engineering

The process of gathering, analysing, documenting, and checking requirements

- Feasibility study
- Requirements elicitation
- Requirements specification
- Requirements validation

Feasibility Study

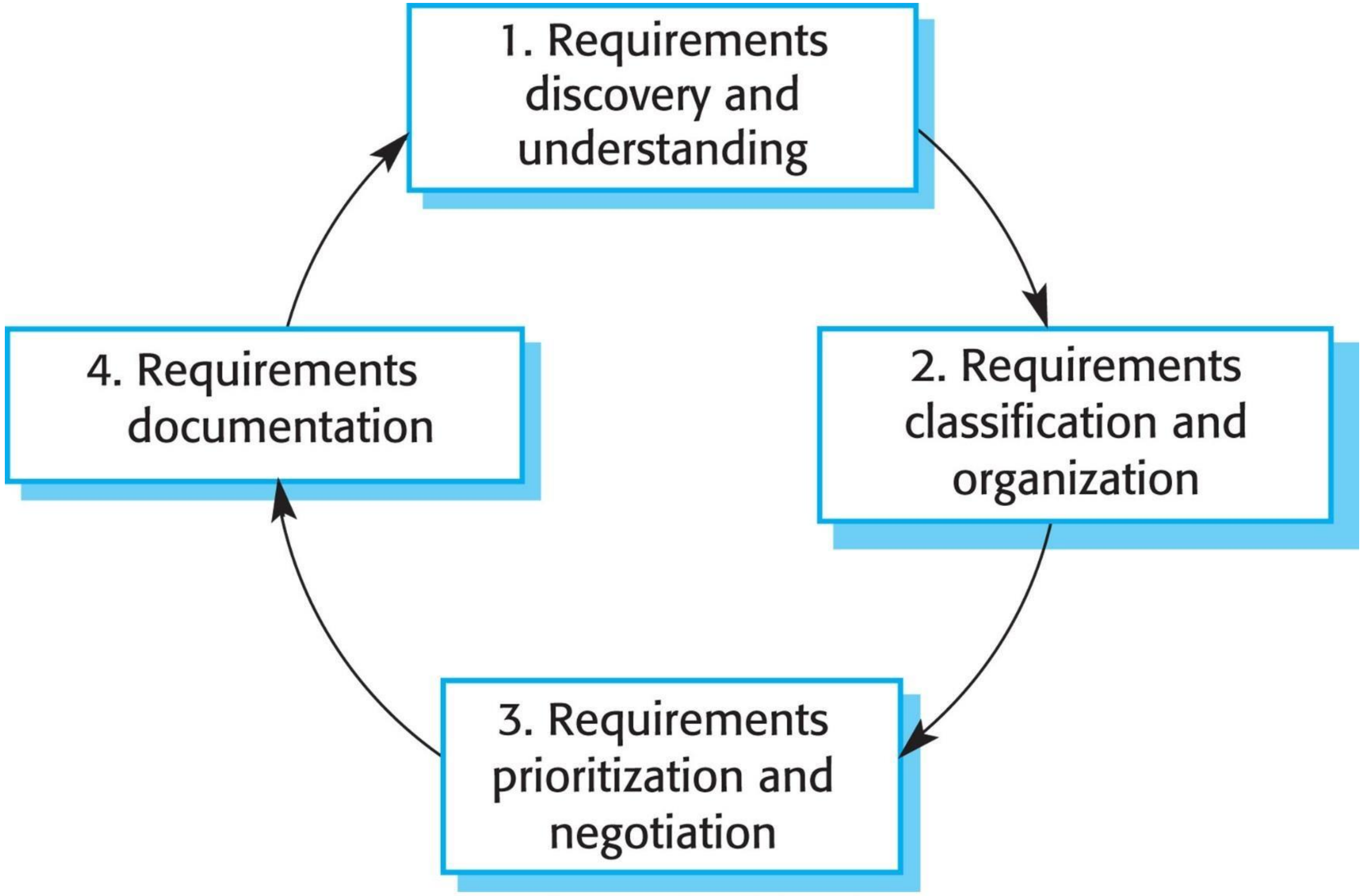
1. Does the system contribute to the overall objectives?
2. Can the system be implemented within schedule and budget using current technology?
3. Can the system be integrated with the other systems that are used?

> *Fail fast*: The earlier failed projects are recognised the better.

How do you get the requirements?

> Erm, yes how do you do it?

Requirements Elicitation


1. Requirements discovery and understanding
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements documentation

Requirements Elicitation Methods

- Interviews & Focus Groups
- Scenarios
- Use cases (Not covered here)
- Ethnography

Interviews

Formal / informal interviews

- Closed (set of questions)
- Open (explore issues)

> Understand what stakeholders do, and the **challenges** they face

- Business process descriptions
- Existing systems
- User observations

Interview Process - Design

- Themes covered
- Participants involved – identify participants' profile
    - Consent form
    - Recording technique

Interview Process - Prepare

- Questions to ask – cover all themes
- Recruit participants – match the profile
- Schedule the interviews

Interview Process - Run

- Record discussion
- Take notes

Interview Process - Transcribe

- Notes
- Full transcript

Interview Process - Analyse

- Identify patterns across responses
- Identify critical responses

Interview Process - Discuss

> Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.

Focus Groups

Interviews held with groups of stakeholders

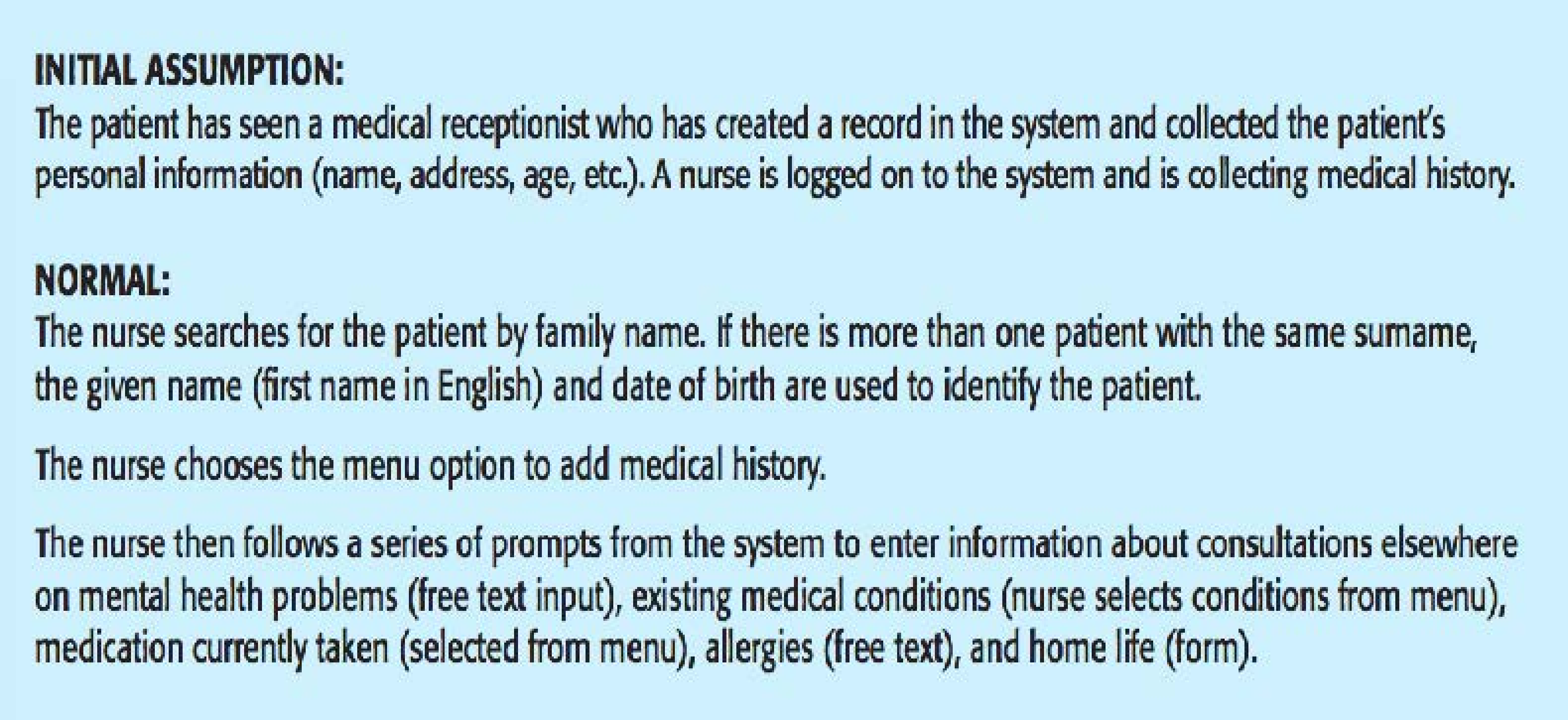
Scenarios

Scenario-based elicitation involves working with users to identify scenarios and to capture:

- What the system and users expects when the scenario starts.
- The normal flow of events.
- What can go wrong and how this is handled.
- Activities that might be going on at the same time.
- The system state when the scenario finishes.

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

Scenario example



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.

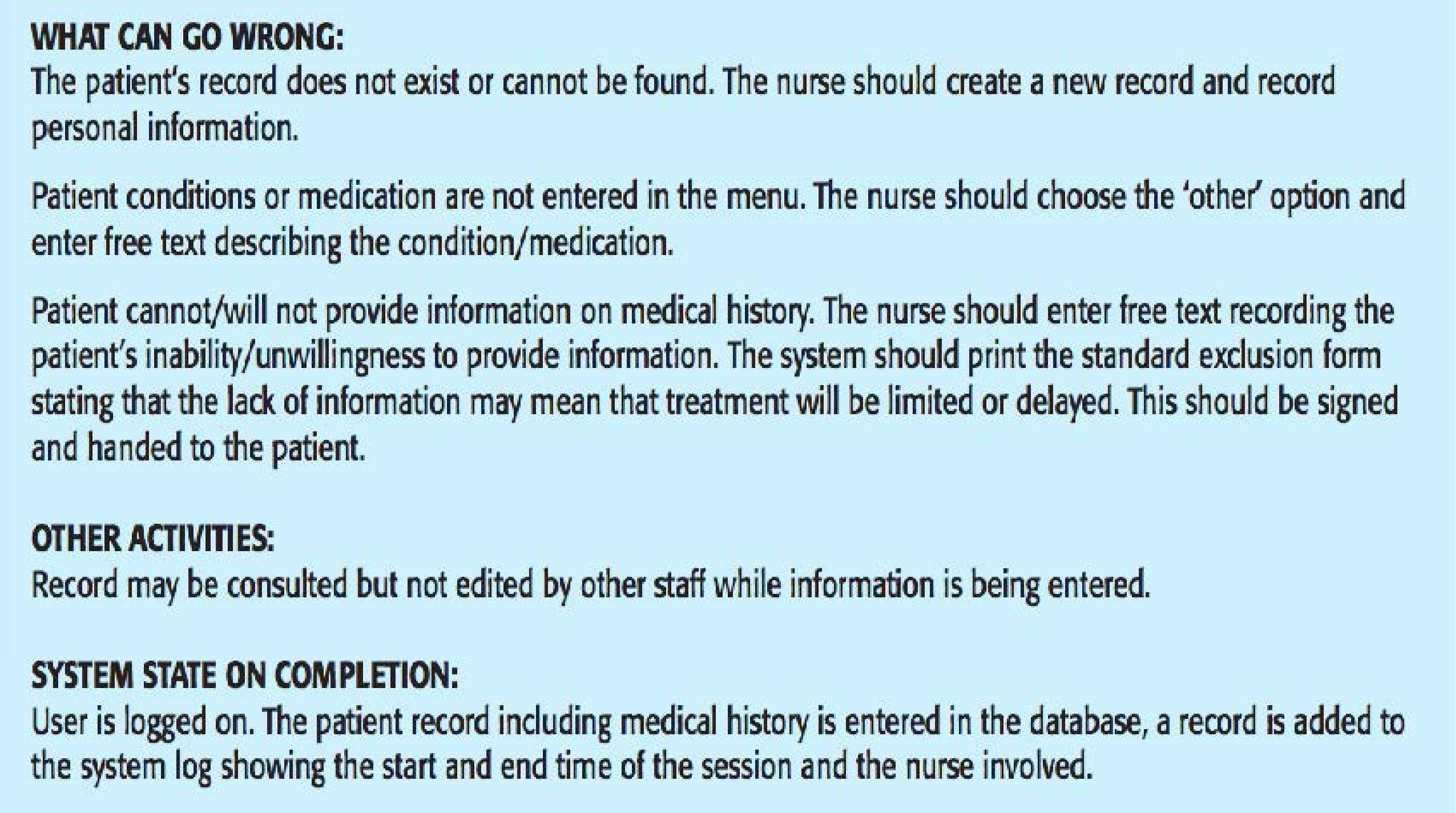
NORMAL:

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).

Scenario example



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.

Patient conditions or medication are not entered in the menu. The nurse should choose the 'other' option and enter free text describing the condition/medication.

Patient cannot/will not provide information on medical history. The nurse should enter free text recording the patient's inability/unwillingness to provide information. The system should print the standard exclusion form stating that the lack of information may mean that treatment will be limited or delayed. This should be signed and handed to the patient.

OTHER ACTIVITIES:

Record may be consulted but not edited by other staff while information is being entered.

SYSTEM STATE ON COMPLETION:

User is logged on. The patient record including medical history is entered in the database, a record is added to the system log showing the start and end time of the session and the nurse involved.

Ethnography

- Observational technique
- Used to understand operational processes and help derive requirements for these processes
- Helps discover implicit system requirements
  - the actual way people work rather than the formal processes defined by the organization
  - people do not have to explain or articulate their work
  - Social and organisational factors of importance may be observed.
  - Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

Ethnography

The ways in which people actually work

- Reality may differ from procedure

Cooperation and awareness of other’s activity

- Change your process based on the workload of your colleagues

Requirements Elicitation Challenges

Scope

- Boundaries of the system are ill-defined
- Unnecessary information is given

Volatility

- Requirements evolve over time

Understanding

- Users have incomplete understanding of their needs
- Users overestimate the available technology and its capabilities
- Analysts have poor knowledge of problem domain

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.

The software requirements specification

- The software requirements specification (SRS) 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
    - what the system should do not how it should do it.

Questions?