

Software Engineering

Requirements Engineering
Tom Verhoeff

1

General Questions

- Why to do it?
- What to do?
- How to do it ?

2

What ? SMART: Mannion et al.

- S = **Specific**
- M = **Measurable** = Verifiable
- A = **Attainable** = Possible at All
- R = **Realizable** = Within Project Constraints
- T = **Traceable**

3

How ? Elicitation, Analysis

- **Interview**, survey (prepare open questions)
- Study **existing products, system requirements**
- Consolidate and structure raw results
- Follow-up, discuss draft requirements
- **Prototype** (could be paper mock-up)

4

General Requirements

- ⑥ Name, purpose
- ⑥ Overview of capabilities, constraints, why's
- ⑥ User categories, typical usage
- ⑥ Operational environment, context diagram
- ⑥ External interfaces

5

Specific Requirements

- ⑥ Atomic (single sentence); identifier, priority
- ⑥ Functional requirements: include capacity, speed, accuracy
- ⑥ Redundancy: good and bad; cross-reference
- ⑥ To Be Defined (TBD), To Be Confirmed (TBC)
- ⑥ About problem domain, NOT solution domain

6

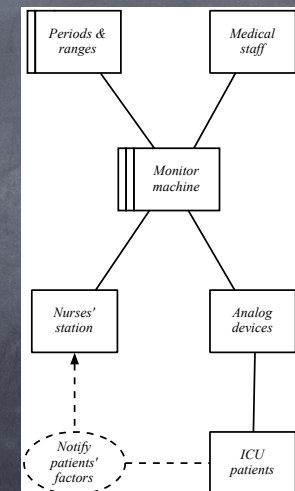
UR Example

A **patient monitoring program** is required for the ICU in a hospital. Each **patient** is monitored by an **analog device** which measures factors such as pulse, temperature, blood pressure, and skin resistance. The program reads these factors on a **periodic** basis (specified for each patient) and stores the factors in a database. For each patient, **safe ranges** for each factor are also specified by **medical staff**. **If** a factor falls outside a patient's safe range, or if an analog device fails, the **nurses' station** is notified.

7

Problem Frames

- ⑥ "Machine" domain
- ⑥ "Designed" domains
- ⑥ "Given" domains
- ⑥ Required relationships



8

SE Principles

- Rigor & formality
- Separation of concerns
- Modularity
- Abstraction
- Anticipation of change
- Generality
- Incrementality