CMPT 276 Class 09: Requirements Engineering

Dr. Jack Thomas

Simon Fraser University

Fall 2020

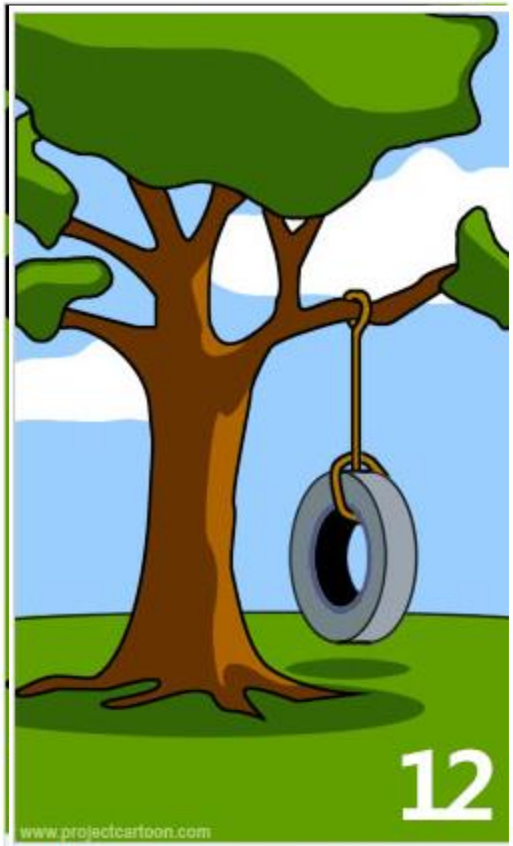
Today’s Topics

1. How are requirements stated for **the customer** vs for **the developer**?

2. What is the difference between **functional** and **non-functional** requirements?

Requirements Engineering

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



What the customer really needed

Two Types of Requirements

- **User requirements**
  - High-level description of what the customer needs the system to do (**qualitative**).
  - Uses English statements and diagrams.
  - May be basis for bidding on a project.
- **System requirements**
  - Document detailing precisely what should be implemented (**quantitative**).
  - Often more formal and technical than the user requirements.
  - May be part of a contract for developing system.

User and System Requirements

- **Example User Requirement**: "The Medical Health Care Patient Management System shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month."
- **Example System Requirement**:
  1. On the last working day of each month, a summary of the drugs prescribed, their cost, and the prescribing clinics shall be generated.
  2. The system shall automatically generate the report for printing after 17:30 on the last working day of the month.
  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.
  4. If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
  5. Access to all cost reports shall be restricted to authorized users listed on a management access control list.

Exercise: User or System Requirement?

- Classify each requirement for a **slide-presentation software** as a **User** or **System** requirement:
  1. A user drags a slide while reordering a slide-deck, other slides move out of the way with an animation effect lasting 0.25s.
  2. System supports reordering slides from a thumbnail view via drag-and-drop.
  3. The user may animate content onto the slide where it is initially hidden and then appears.
  4. Animations for the current slide are displayed in a tree view.
  5. Animations may be reordered using up and down arrows at the bottom of the display window.

Functional vs. Non-Functional Requirements

- **Functional Requirements**
  - What functions the system should provide.
    - Ex: How the system should react to particular inputs or particular situations.
  - May state what the system should **not** do.
- **Non-Functional Requirements**
  - Constraints on the system such as timing, development process, or standard compliance.
  - Often applies to the whole system rather than individual features.

Functional Requirements

- **Functional user requirements**:
  - High-level statements of what the system should do.
- **Functional system requirements**:
  - Describe the system services in detail.
- Problems arise when requirements are not precisely stated.
  - **Ambiguous requirements** may be interpreted differently by developers and users:
    - “Police help dog bite victim”
    - “Kids make nutritious snacks”
    - “One morning I shot an elephant in my pyjamas”

Requirements & Imprecision

- Functional requirements for the MHC-PMS
  - A user shall be able to search the appointments lists for all clinics.
  - Each staff member shall be uniquely identified by his or her 8-digit employee number.
- Consider the term ‘search’:
  - User intention:
    - Search for a patient across all appointments in all clinics.
  - Developer interpretation:
    - Search for a patient in any one specific clinic.

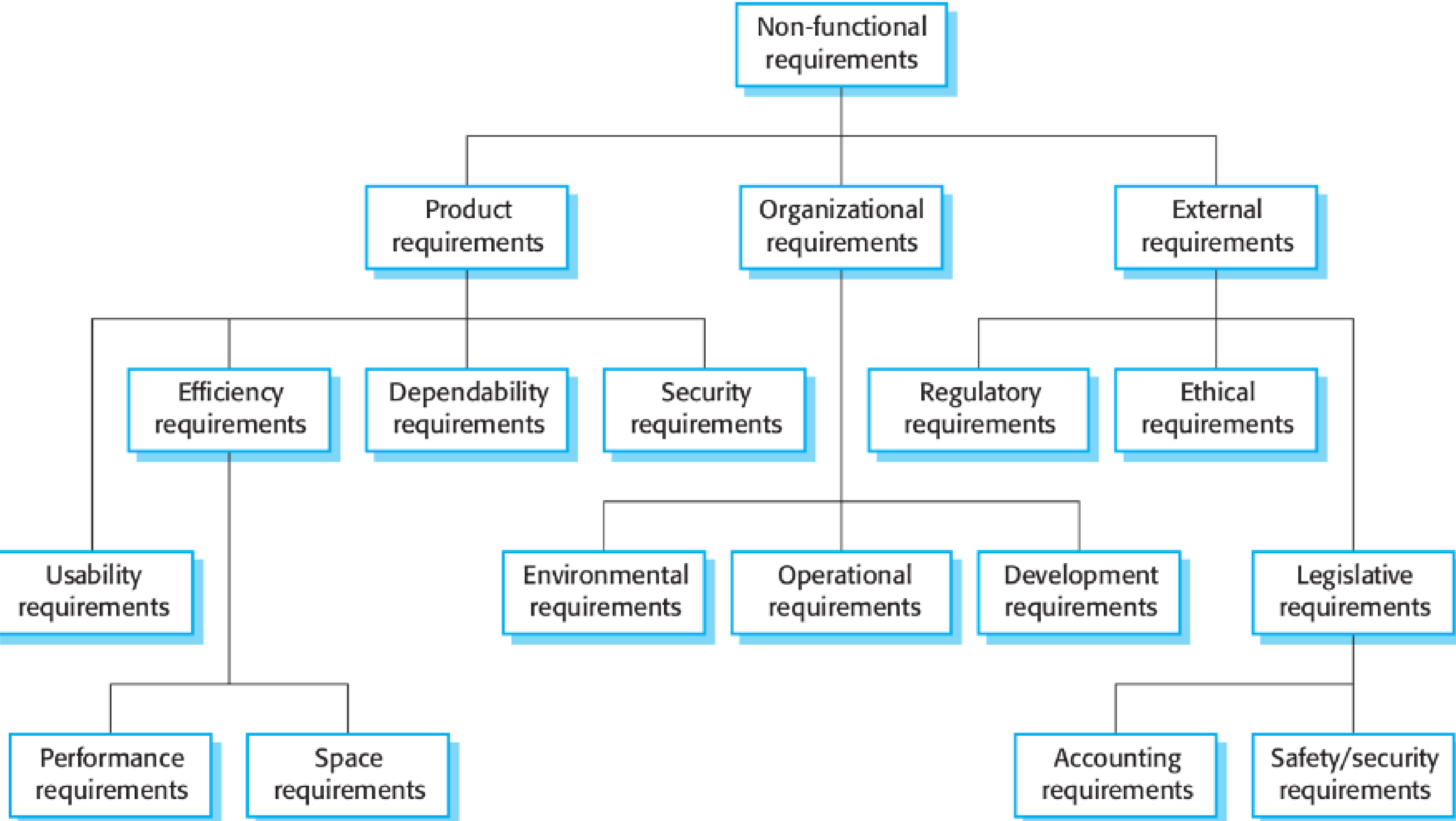
Requirements Completeness and Consistency

- In principle, requirements should be both **complete** and **consistent**.
  - **Complete**: Describes all required functionality
  - **Consistent**: No conflicts or contradictions in the requirements.
- **Practically impossible** to produce a complete and consistent requirements document.

Non-functional Requirements

- Non-Functional requirements define:
  - **System-level properties and constraints**: reliability, response time and storage space.
  - **Developing process constraints**: programming language or development method.
- Non-functional requirements **may be more critical** than functional requirements.
  - If they are not met, the system may be useless.

Types of Non-Functional Requirements



Non-Functional Requirements Implementation

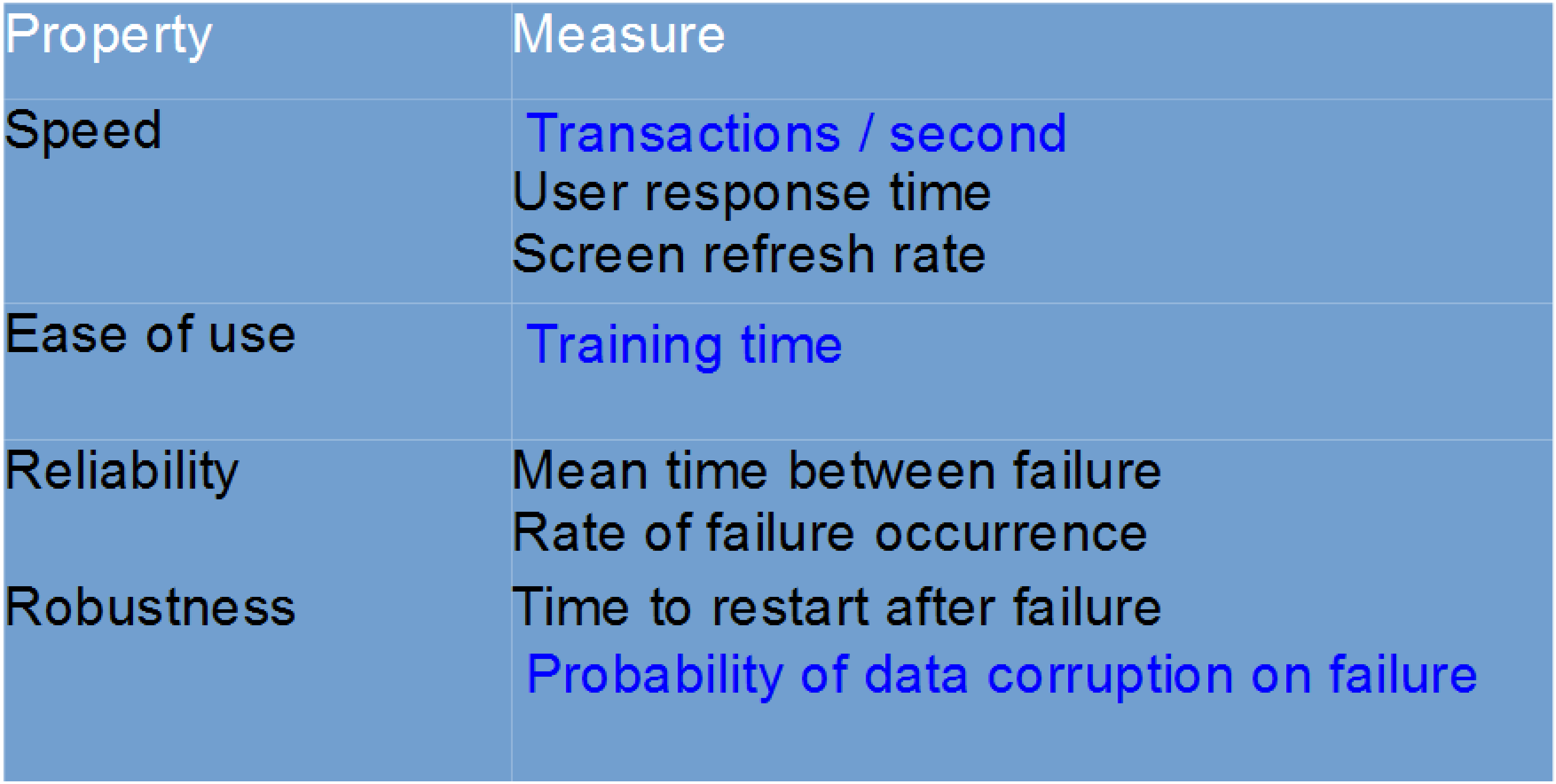
- Non-functional requirements **may affect the overall system architecture** (rather than single components).
  - **Ex**: organize system to minimize communication to meet performance requirements.

- A non-functional requirement may **impose** a number of **functional requirements**.
  - **Ex**: security needs may dictate numerous features to meet those needs.

Quantitative Non-Functional Requirements

- Write non-functional requirements **quantitatively**:
  - **User Goal**: "The system should be easy to use and organized such that user errors are minimized."
  - **Verifiable Requirement**: "After 4 hours of training, average user error shall be less than 2 per hour."

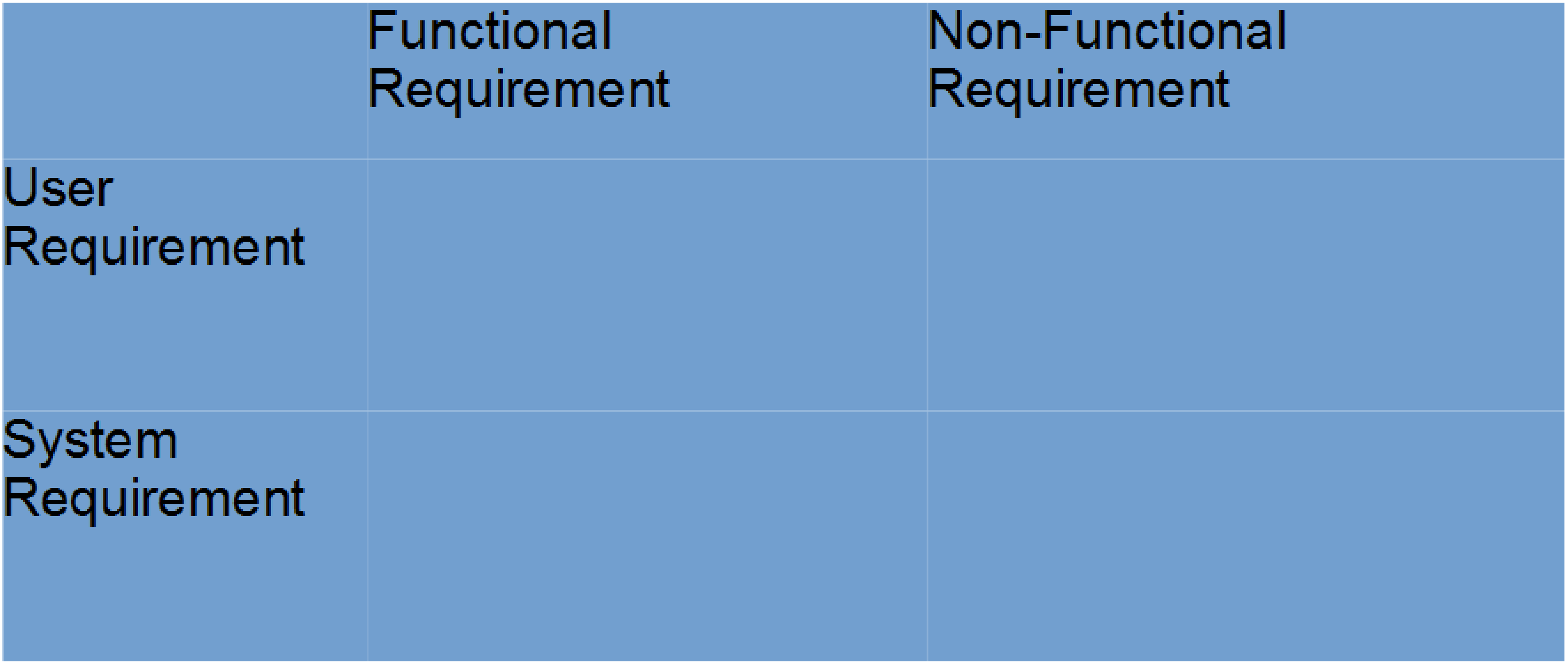
Metrics for Specifying Non-Functional Requirements.



Property	Measure
Speed	Transactions / second User response time Screen refresh rate
Ease of use	Training time
Reliability	Mean time between failure Rate of failure occurrence
Robustness	Time to restart after failure Probability of data corruption on failure

Class Exercise

- Let’s fill the grid with requirements for the minesweeper game from assignment 3.



	Functional Requirement	Non-Functional Requirement
User Requirement		
System Requirement		

Recap – The Required Reading

- **Requirements** define
  1. What **the system should do**.
  2. **Constraints** on its operation and implementation.
- **Functional** requirements:
  - – The services that the system must provide.
- **Non-functional** requirements:
  - – Constrain the system or development process.
  - – Often relate to emergent properties of the system.
  - – Apply to the system as a whole.