CISTER - Research Center in
Real-Time & Embedded Computing Systems
New Frontiers
in Computing

RAMDE

Requirements and Model-Driven Engineering

MScCCSE 2021/22

CISTER – Research Centre in
Real-Time & Embedded Computing Systems

RAMDE's motto

Specifying and modelling systems to:

1. guide the engineer to build
2. guide the engineer to collaborate
3. reason precisely about it

- The 3rd part is often too expensive – in terms of time and effort.

In this course, we try to bridge the gap between:

engineering models and formal methods

Important Remark: **Critical Computing Systems demand rigor, thus the need for mathematical support via formal methods**

Syllabus

- High-level overview or requirements and associated processes
- Mathematical Preliminaries
  - Basic mathematical notations
  - Set theory
  - Propositional Logic
    - Syntax, semantics, and reasoning
  - First Order Logic
    - Syntax, semantics, and reasoning
  - The Z3 automatic theorem prover
    - Rise4fun interface: get acquainted with the tool
    - Python API: automating search for solutions

- Behavioural modelling
  - Single component
    - State diagrams and Flow charts
    - Formal modelling: Automata, Process Algebra and equivalences
  - Many components
    - Communication diagrams and Sequence diagrams
    - Formal modelling: Process algebra with interactions, Realisability
    - Verification of requirements
    - Formal modelling: modal logics
  - Tools: model checking with mCRL2

Evaluation

Individual exercises (10%)

- Select exercises from exercises sheet
- Feedback during the classes
- Submit a pdf with the answer within 2 weeks after exercises sheet is released

Individual exercises (60%)

- 2 assignments
- Groups of 4 students each
- Feedback during classes
- Submissions via git

Literature Review (30%)

- 1 scientific paper per group
- Report: summary + critics
- Oral presentation
- Peer-evaluation among groups

More information

https://cister-labs.github.io/ramde2122

- These slides
- Assignments
- Useful links
- Bibliography
- More detailed rules

Collaboration

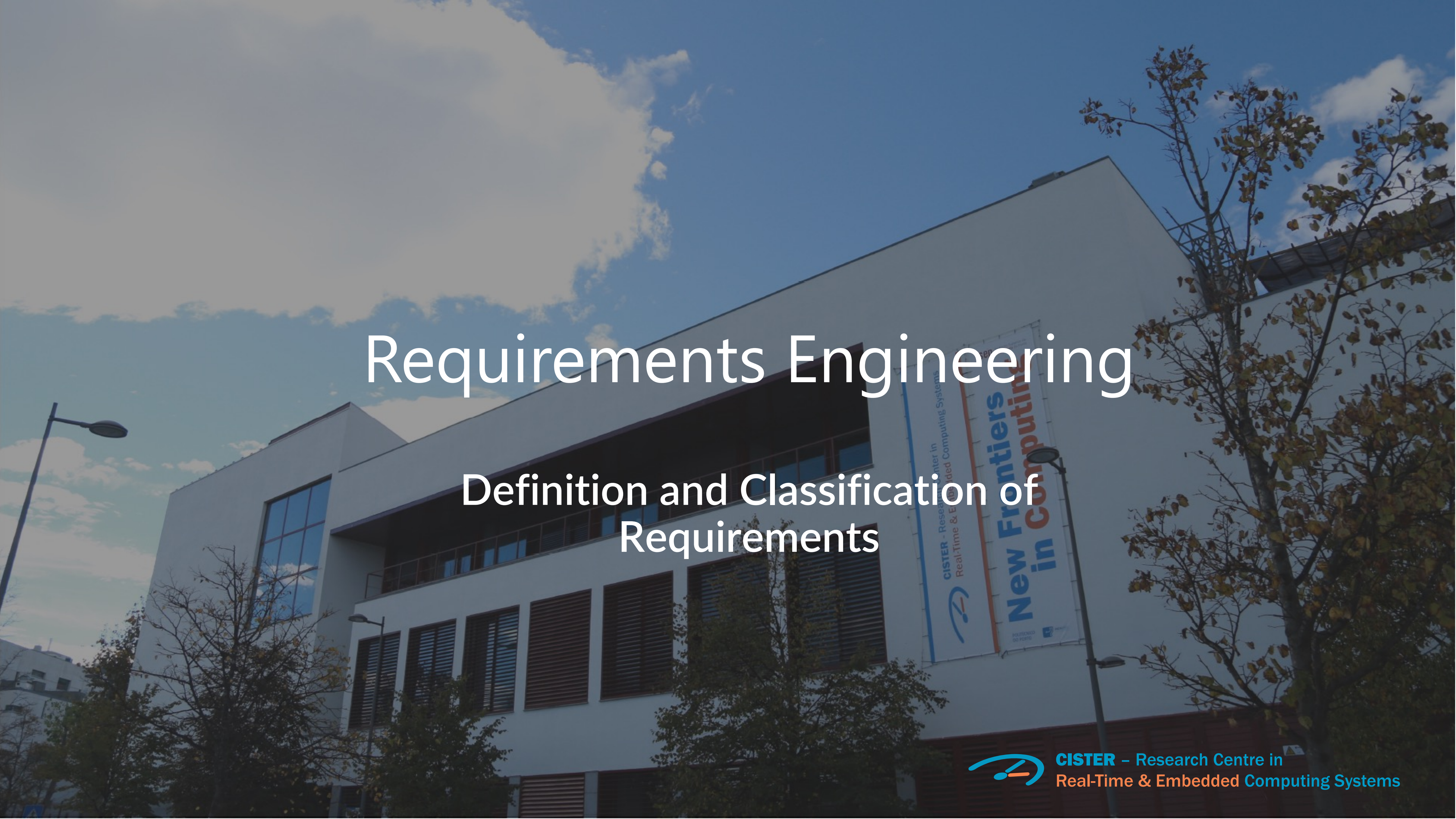
- MS teams to communicate, present, share.
- MS teams to ask lecturers or schedule virtual meetings
- GitHub to host repositories for submissions
- Moodle kept at minimum

For today

- An high-level overview of:
  - Requirements
  - Specifications
  - Models

RAMDE's approach

How to handle these in a rigorously and precisely in such a way that we are able to reason about systems


CISTER - Research Center in
Real-Time & Embedded Computing Systems
New Frontiers
in Computing

Requirements Engineering

Definition and Classification of Requirements

CISTER – Research Centre in
Real-Time & Embedded Computing Systems

But, what is a requirement?

IEEE 729-1983 - *IEEE Standard Glossary of Software Engineering Terminology*

- A condition or capability needed by a user to solve a problem or achieve an objective

- A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents

- A documented representation of a condition or capability as defined by the two previous points

Types of requirements

- In this class, we will be looking at:
  - Functional
  - Non-Functional
  - Domain

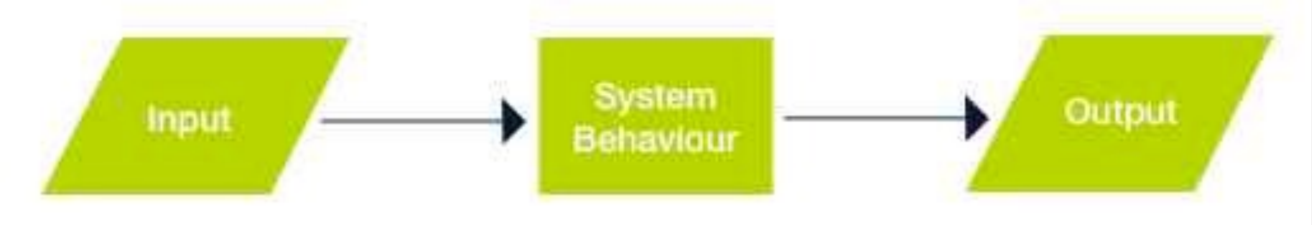
Types of requirements

Functional Requirement Definition

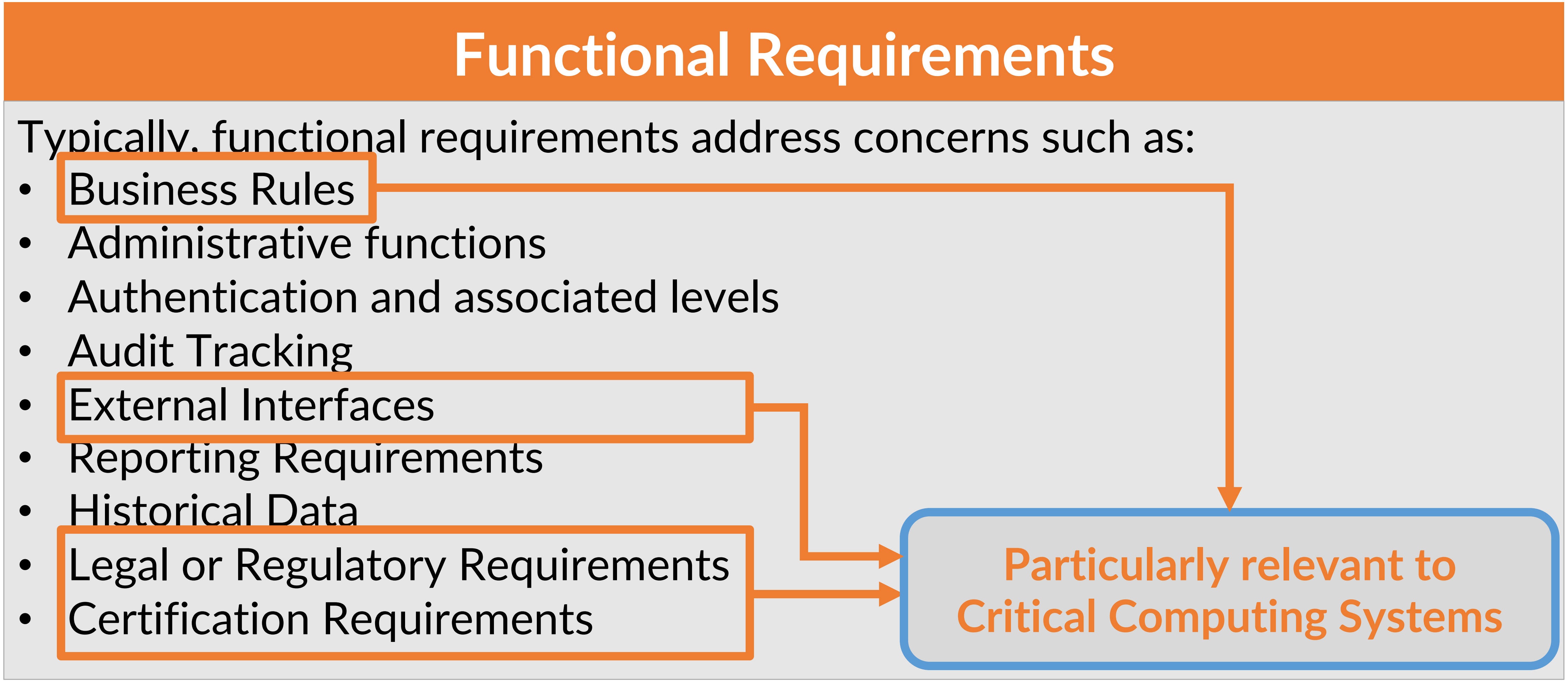
- Are the requirements that the **end user specifically demands** as **basic facilities** that **the system should offer**, and that **must be part of the contract**. Putting it in the simplest way:

  "Any Requirement Which Specifies What The System Should Do."

- These are **represented** or **stated** in the form of **input to be given** to the system, the **operation performed**, and the **output expected**. In a nutshell:



Types of requirements



Functional Requirements

Typically, functional requirements address concerns such as:

- Business Rules
- Administrative functions
- Authentication and associated levels
- Audit Tracking
- External Interfaces
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements
- Certification Requirements

Particularly relevant to Critical Computing Systems

Types of requirements

Non-Functional Requirement Definition

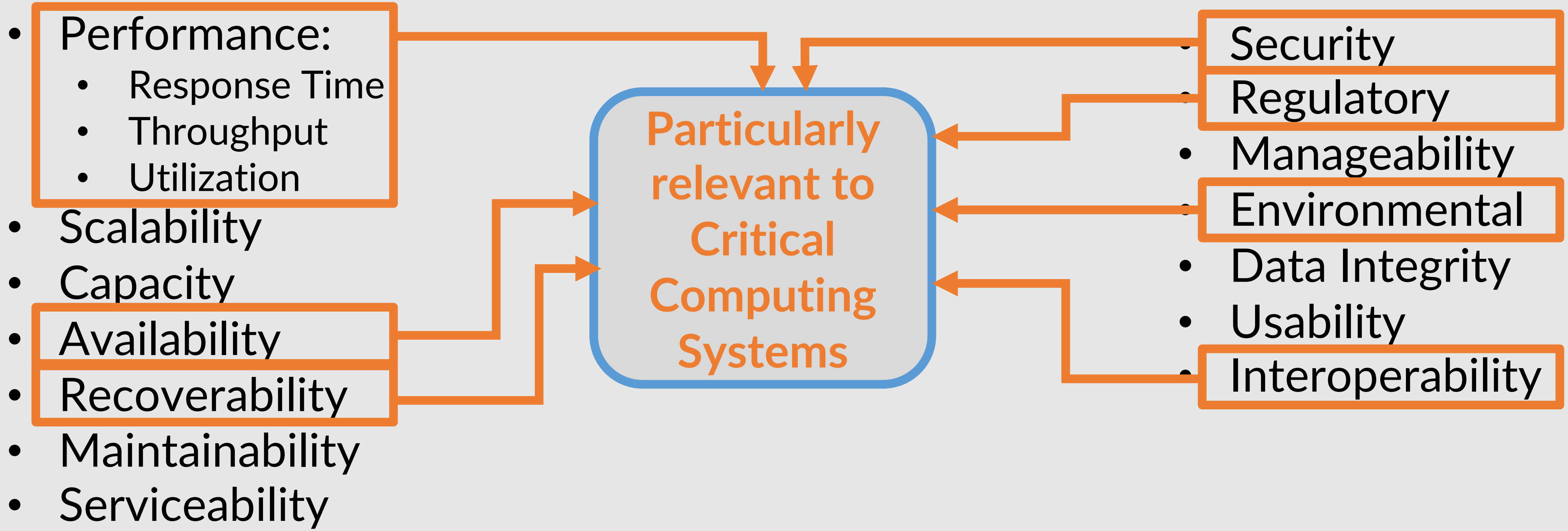
- The quality constraints that the system must satisfy according to the project contract.
- Also known as "non-behavioral requirements"

"Any Requirement That Specifies How The System Performs A Certain Function."

Types of requirements

Non-Functional Requirements

Typically, functional requirements address concerns such as:



- Performance:
  - Response Time
  - Throughput
  - Utilization
- Scalability
- Capacity
- Availability
- Recoverability
- Maintainability
- Serviceability

Particularly relevant to Critical Computing Systems

- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

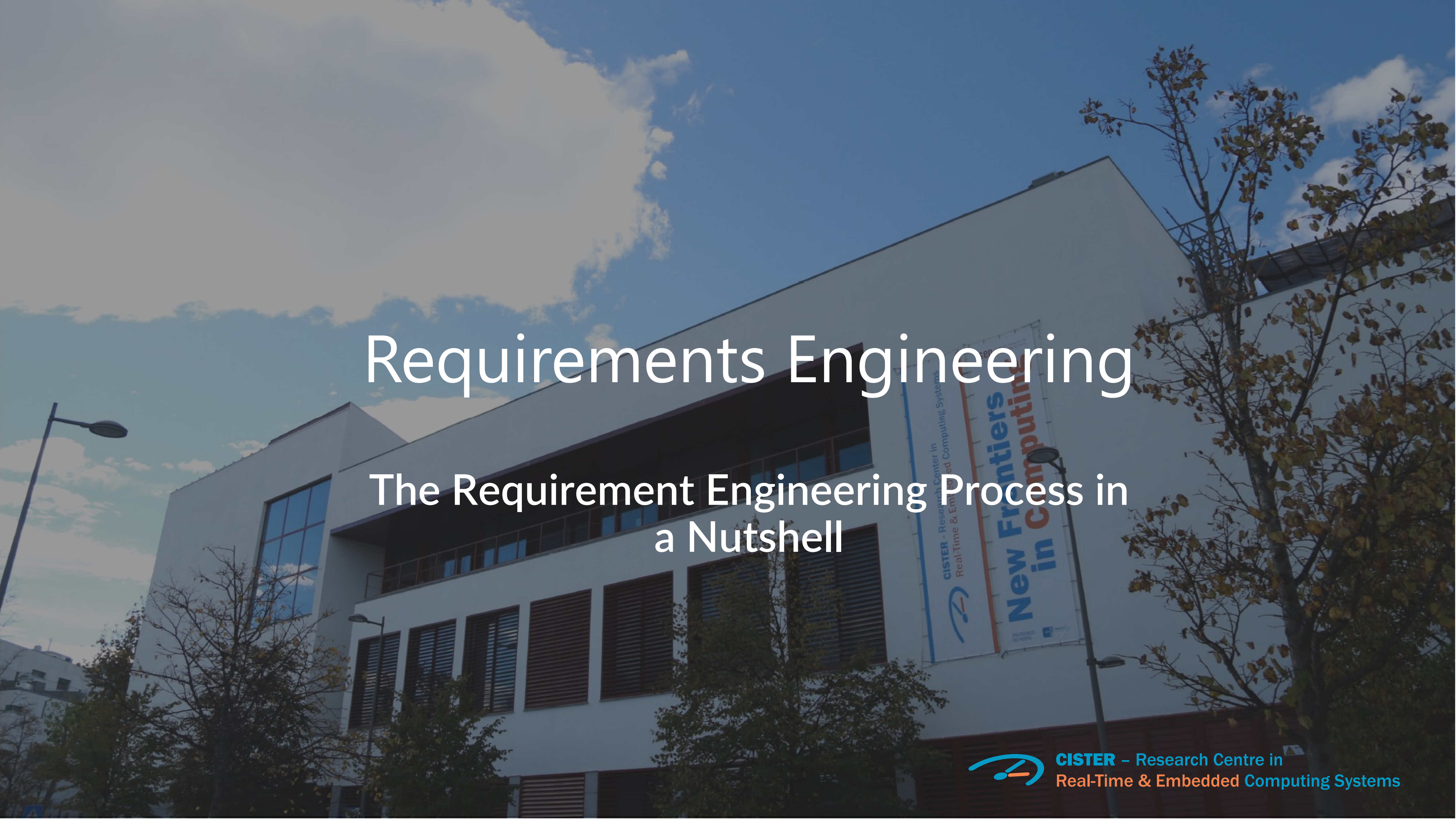
Types of requirements

Domain Requirement Definition

- Requirements that are specific and basic functions of the application domain of the system.
- Example:

  "a train control system has to take into account the braking characteristics in different weather conditions"

- How to define domain requirements?
  - as new functional requirements
  - as new constraints on existing requirements


CISTER - Research Center in
Real-Time & Embedded Computing Systems
New Frontiers
in Computing

Requirements Engineering

The Requirement Engineering Process in a Nutshell

CISTER – Research Centre in
Real-Time & Embedded Computing Systems

Requirement Engineering Process: Overview

Requirements Engineering Process in a nutshell

- Is the process of defining, documenting and maintaining the requirements.
- It is a process of gathering and defining service provided by the system.

Activities Involved

- Requirements Elicitation
- Requirements Specification
- Requirements Verification & Validation
- Requirements Management

Requirements Elicitation Process

What is Requirements Elicitation

- The set of activities used to obtain knowledge about the system's domain and requirements.

- The activities focus on understanding:
  - the application domain where the system will operate
  - the details of the precise problem where the system will be applied
  - how the system interacts with external business requirements
  - the needs and constraints of the stakeholders

Requirements Elicitation Process

Requirements Elicitation Stages

- **Objective Setting**:
  - establish objectives, define general goals of the business, outline description of the problem, identify the system's constraints
- **Background knowledge acquisition**:
  - information about the environment where the system will be deployed/installed, the application domain, and other systems
- **Knowledge organisation**:
  - information collected during background acquisition, must be organized
- **Stakeholder requirements collection**:
  - interacting with stakeholders to identify their needs and constraints

Requirements Elicitation Process

Analysis Checks

- **Necessity checking**:
  - analyse the **need** for the requirement (*e.g., requirement not contributing to the business goals and/or to the specific problem of the system*)
- **Consistency and completeness checking**:
  - requirements are cross-checked for **consistency** (*no requirements are contradictory*) and **completeness** (*no services or constraints which are needed have been missed out*)
- **Feasibility checking**:
  - requirements are checked for their **feasibility** considering the available **schedule** and **budget** for developing the system

Requirements Elicitation Process

Negotiation

- **Requirements discussion**:
  - problematic requirements are discussed, and the stakeholders involved present their views
- **Requirements prioritisation**:
  - requirements are prioritised to identify critical requirements (*helping the decision-making process*)
- **Requirements agreement**:
  - solutions to the identified problems are identified, and a compromise set of requirements are agreed (*possibly*, *leading to changes on existing ones*)

Requirements Elicitation Process

Techniques

- **Interviews:**
  - the requirements engineer/analyst addresses the several **stakeholders** to build up an **understanding** of their requirements (*closed* vs. *open*)
- **Scenarios:**
  - **stories** which **explain** how a system might be **used** (*system state before entering the scenario, flow of events, exceptions, concurrent activities, system state after end of scenario*)
- **Requirements reuse**:
  - taking the requirements which have been developed for **one system** and using them in a **different system** (reduce time and effort)

Requirements Elicitation Process

Techniques

- **Prototyping:**
  - A prototype is an initial version of a system which may be used for experimentation
    - Establishes feasibility and usefulness before high development costs are incurred
    - allows users to experiment and discover what they really need
- **Throw-away prototyping** (*help elicit and develop the system requirements*) vs. **Evolutionary prototyping** (*deliver a workable system quickly to the customer*)

Requirements Specification

What is Requirement Specification

- The goal is to produce formal software requirement models.
- During specification, more knowledge about the problem may be required which can again trigger the elicitation process.
- **A lot more to come on requirement specification and models, latter on**

Requirements Verification & Validation

What is Requirements V&V

- **Verification:** set of tasks that ensures that the software **correctly implements** a specific function (*closer to code, e.g., unit testing, formal verification, simulation, etc.*).
- **Validation:** set of tasks that ensures that the software that has been built **is traceable** to customer **requirements**.
  - several checks may be required (*Completeness, Consistency, Validity, Realism, Ambiguity, Verifiability, etc.)*
- The output of requirements V&V is the list of **problems** and the agreed **actions** that should be taken to **fix the detected problem**.

Requirements Management

What is Requirements Management

- Is the process of **analyzing**, **documenting**, **tracking**, **prioritizing** and **agreeing** on the requirement and **controlling the communication** to relevant stakeholders.
- This stage takes care of the **changing nature** of requirements. It should be ensured that the requirement specification is as **modifiable** as possible so as to **incorporate changes** in requirements specified by the end users (*possibly at later stages too*).

isep
New frontiers in computing
CISTER Research Center in Real-Time & Embedded Computing Systems

The End

We will continue with a formally driven view of requirement, specifications, and models…