Politecnico
di Milano

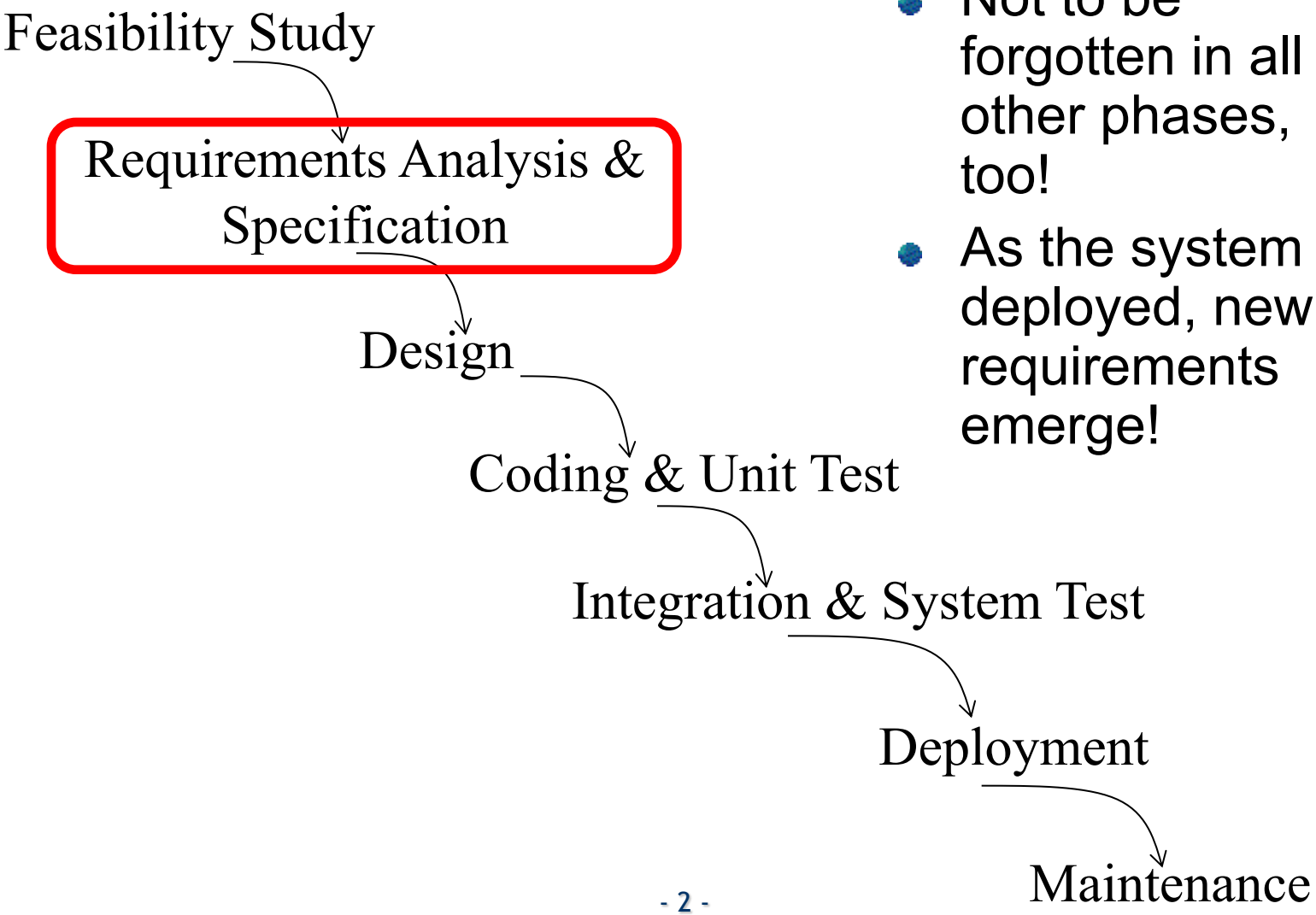
Requirements engineering (RE)

Context

Definitions

Importance and difficulties

Context: where do we find RE?



- Not to be forgotten in all other phases, too!
- As the system is deployed, new requirements emerge!

Requirements engineering: definition

- [Nuseibeh&Easterbrook ’00]
  - The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended
  - Software systems requirements engineering (RE) is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation
    - Important issues
      - *Identify stakeholders*
      - *Identify their needs*
      - *Produce documentation*
      - *Analyse, communicate, implement requirements*

Requirement engineering: definition

- From Zave ‘83
  - Requirements engineering is the branch of software engineering concerned with the
    - real-world goals for,
    - functions of, and
    - constraints on
  - software systems!
  - It is also concerned with the relationship of these factors to **precise specifications of software behaviour, and to their evolution over time and across software families**

What is a requirement?

- Examples of candidate requirements
  - “The system shall allow users to reserve taxis”
  - “The system has to provide a feedback in 5 seconds”
  - “The system should never allow non-registered users to see the list of other users willing to share a taxi”
  - “The system should be available 24/7”
  - “The system should guarantee that the reserved taxi picks the user up”
  - “The system should be implemented in Java”
  - “The search for the available taxi should be implemented in class Controller”

Functional and non-functional requirements

Types of requirements

- **Functional** requirements:
  - Describe the interactions between the system and its environment independent from implementation
  - Examples:
    - A word processor user should be able to search for strings in the text
    - “The system shall allow users to reserve taxis”
  - Are the main goals the software to be has to fulfill
- **Nonfunctional** requirements:
  - User visible aspects of the system not directly related to functional behavior
  - Examples:
    - The response time must be less than 1 second
    - The server must be available 24 hours a day
- **Constraints** (“pseudo requirements”):
  - Imposed by the client or the environment in which the system operates
    - The implementation language must be Java
    - The credit card payment system must be able to be dynamically invoked by other systems relying on it

Characteristics of nonfunctional requirements

- Constraints on how functionality has to be provided to the end user
- Independent of the application domain
- … but the application domain determines
  - Their relevance
  - Their prioritization
- Have a strong impact on the structure of the system to be
  - Example: if a system has to guarantee to be available 24 hours per day, it is likely to be thought as a (at least partially) replicated system
- Also called **Quality of Service (QoS)** attributes

Some relevant QoS characteristics

- Performance
- Reliability
- Scalability
- Capacity
- Accuracy
- Accessibility
- Availability

Externally visible properties

- Robustness
- Exception handling

How the system works in unexpected/fault conditions

- Interoperability
- ….

Systems developed with different frameworks can work together at run time

- Integrity
- Confidentiality
- …

Security issues

Examples of bad requirements

- The system shall validate and accept credit cards and cashier's checks…high priority
- The system shall process all mouse clicks very fast to ensure users do not have to wait
- The user must have Adobe Acrobat installed

Examples of bad requirements

- The system shall validate and accept credit cards and cashier's checks…high priority
  - **Problem**: two requirements instead of one
  - If the credit card processing works, but the cashier's check validation does not… is this requirement pass or fail? Has to be fail, but that is misleading
  - Maybe only credit cards are high priority and cashier's checks are low priority
- The system shall process all mouse clicks very fast to ensure user's do not have to wait
  - **Problem**: this is not testable…quantify how fast is acceptable?
- The user must have Adobe Acrobat installed
  - **Problem**: this is not something our system must do
  - It could be in the constraints/assumptions or maybe operating environment sections, but it is not a functional requirement of our system

Exercise

- Classify the following requirements in functional, non functional and technological constraints
- Highlight bad requirements
  - “The system shall allow users to reserve taxis”
  - “The system has to provide a feedback in 5 seconds”
  - “The system should never allow non-registered users to see the list of other users willing to share a taxi”
  - “The system should be available 24/7”
  - “The system should guarantee that the reserved taxi picks the user up”
  - “The system should be implemented in Java”
  - “The search for the available taxi should be implemented in class Controller”

Plan for next RE class

- Next RE class will be flipped
- **Before** coming to class we ask you to watch the short video you find here
  - https://polimi365-my.sharepoint.com/:v:/g/personal/10143828_polimi_it/Eb4Mp2L_i6tGqX6_TtFepmwB7e9nytIwrBquqqmVI7sJ8A?e=Gx0gMn
- And to answer to the following questionnaire (three questions that will not be used for your evaluation!)
  - https://forms.gle/easKaHYGBsaeSYxe9