# SE - 2
# Requirement engineering

Lydie du Bousquet
Lydie.du-bousquet@imag.fr

In collaboration with J.-M. Favre, I. Parissis, Ph. Lalanda, Y. Ledru

# Schedule

- What / Who / Why …
  requirement engineering
- Serious game: about requirement elicitation
- Serious game: try to do it
- More about Requirement specification


WHY?
HOW
WHAT?
WHO?
WHEN?
WHERE?


# Who is concerned by requirement engineering?


# People

- The **customer** side

  pays for the product and
  usually decides the requirements
- The **supplier/provider** side

  produces a product for a customer
- The **User**

  operates or interacts directly with the product
  may be different from the customer


- What are requirements?
- What is requirement engineering?

WHAT?

# Need to know the requirements to be able to build the program


# Requirements are **statements** of

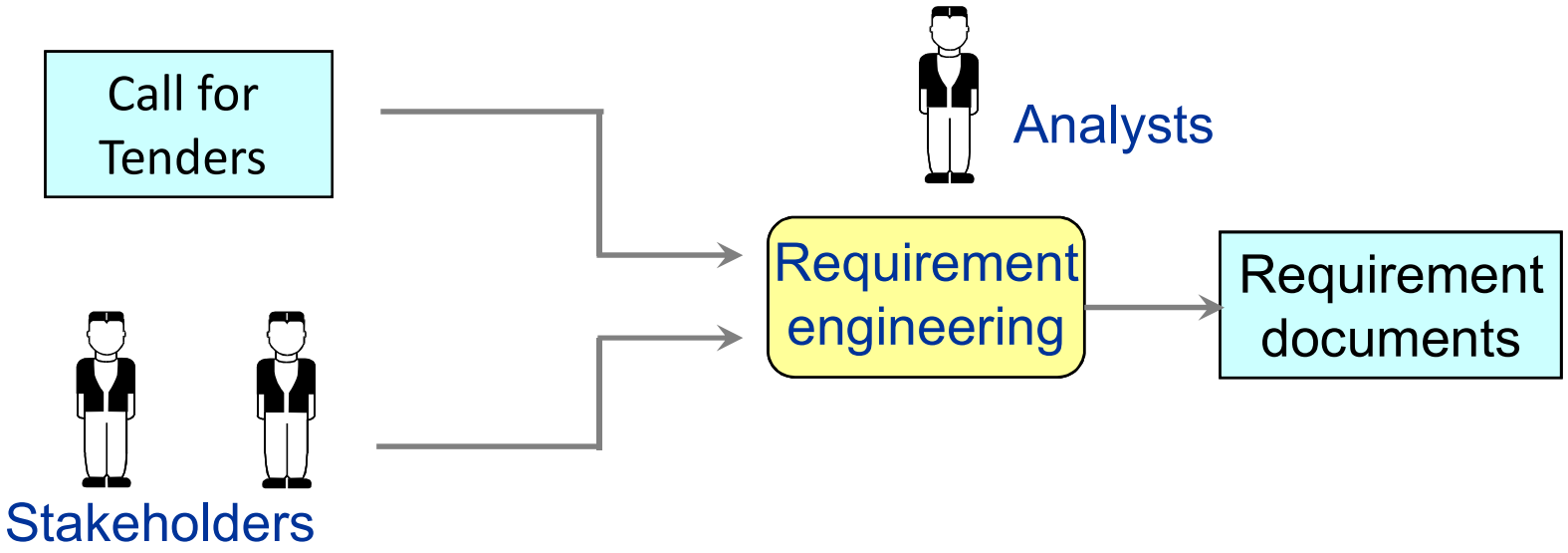
- **what** the system must **do**,
- **how** it must **behave**,
- the **properties** it must exhibit,
- the **qualities** it must possess, and
- the **constraints** that the system and its development must satisfy

WHAT?

# Requirement engineering

## Objectives

- Find out the needs and constraints of the customers
- Specify them in a dedicated document



# Requirement engineering


## Objectives

- **Call for tender**
  - first expression of the customer's **needs** and **constraints**
  - basis for a bid for a contract
- **Stakeholders**
  - All the **people** having an **interest** in the project
  - Customer side: users, experts, managers, sales men ...
  - Supplier side: sales men, development teams, architects, managers, strategist
- **Requirements**
  - Come from the **customer side**
  - Written by customer, supplier, or both in a **Software Requirements Specification** document (SRS)


- Why requirement engineering?


# Requirement Engineering

The **hardest** single part of building
a software system
is deciding **precisely**
**what to build**. . .

WHY?

# Deciding **precisely**
# **what to build is hard**


# Deciding **precisely** **what to build…**

- Customer and supplier may speak **different languages**
- Customer may **not know** precisely what he want
- The needs may **change**
- There may be **conflicts**
- Problem may be **difficult** to be understood
- Many different kinds of information


# Requirement Impacts

- **Legal** impacts
  - Basis of the **contract** between customer and supplier
- **Economic** impacts
  - Cost of **correcting** wrong requirements
  - **Relevance** of the marketed product
- **Social** impacts
  - Wrong requirements may cause **disasters**
- **Usage** impacts
  - **Acceptance** or **rejection** of a software

**How?**

- How is organized Requirement engineering?

# How is organized Requirement engineering?

**How?**

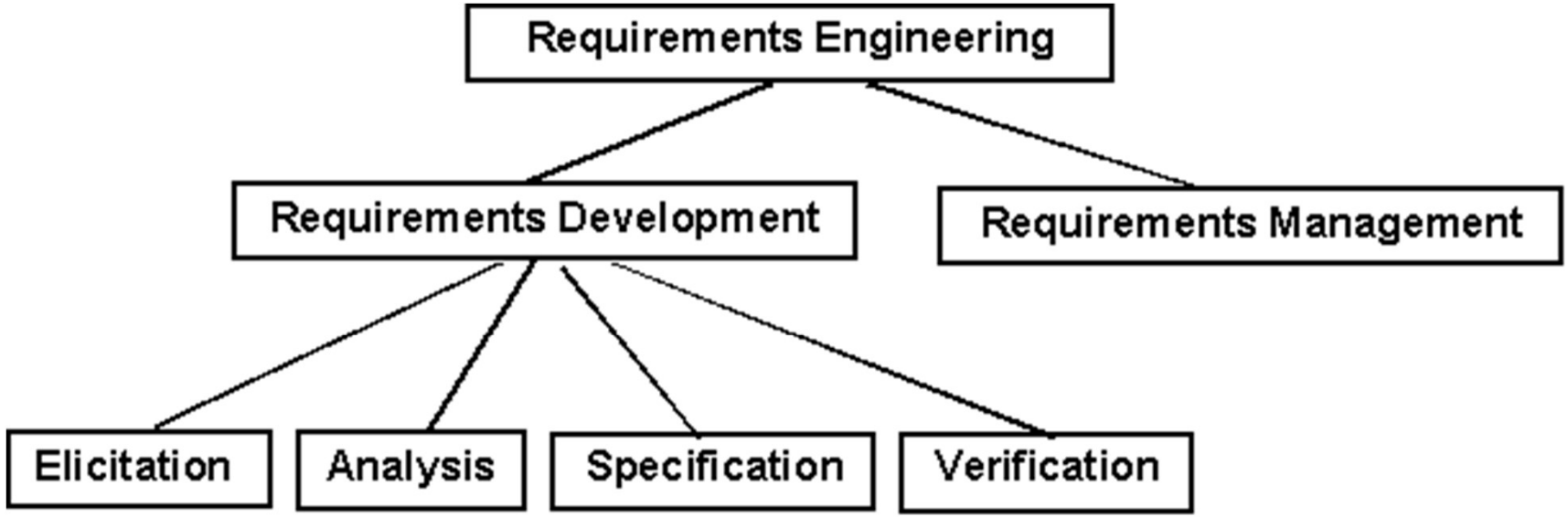
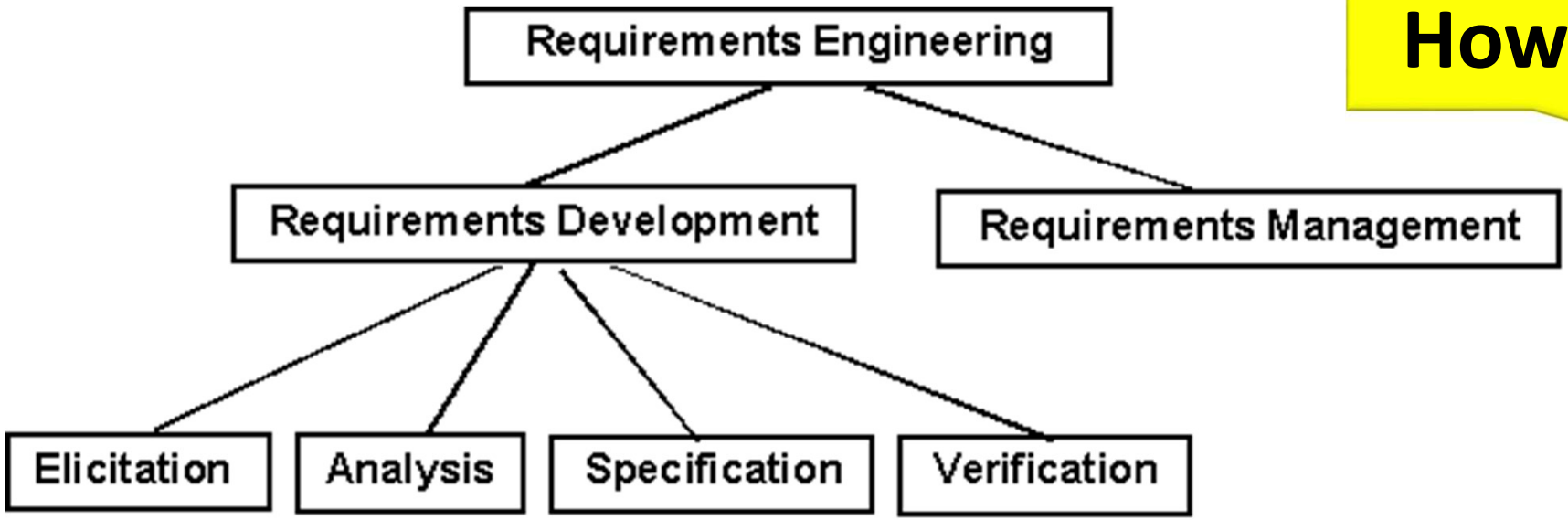



Figure 2. Subdisciplines of requirements engineering.

http://www.processimpact.com/articles/telepathy.html




How?

Figure 2. Subdisciplines of requirements engineering.

How is organized Requirement Engineering?

# REQUIREMENT DEVELOPMENT

Elicitation → Analysis → Spec. → Verif.

# Requirement elicitation

- Learning and understanding the needs of the users
- To avoid confusion between stakeholders and analyst
  - Understand the application domain
  - Identifying the source of requirements
  - Analyzing the stakeholders
  - Selecting techniques, approaches and tool to use
  - Eliciting the requirements
- Types of methods
  - Conversational (based on conversation)
  - Observational (based on observation)
  - Analytic (based on an analysis)
  - Collaborative (needs collaboration of different stakeholders)

# Requirement elicitation -2
# Difficulties

- Users are not fully aware of what they will obtain
- They may not make the difference between
  - What they need and what they have
  - What they want and what they need
- They may not want to work on the problem
- They may use specific language
- They may have forgotten some important information
- The analyst may want to find a solution before knowing the problem or may conclude too quickly

# Requirement elicitation -3
# Example of **communication problem**

- The user requests to change  an incorrect algorithm on the existing system
  - Analyst: “How often this algorithm is used ?”
  - User: “Never”
  - So the request is ignored

- Of course, the reason why the algorithm is not used is because it is **incorrect**!
  - Be careful to rapid conclusion
  - Be careful not to decide for the user

# Requirement elicitation -4
# Example of **communication problem**

- The user “My software is too slow”
- Analyst 1:
  “I think it is due to the hardware”

- Analyst 2:
  “Could you tell me why you think it is slow?”

Who do you trust more? Why?

# Requirement analysis

- Analyze the results of elicitation
  - are the answers consistent?
  - identify trouble spots/conflicts
  - identify limits?
  - identify most important requirements?
- Possibly iterate over elicitation again
- Conflict resolution

# Requirement specification

- Process of writing down the requirements
- No standard nor methods
- From informal to formal
- Functional and non-functional

- **Software Requirements Specification** document (SRS)

Detailed after

# Requirement verification

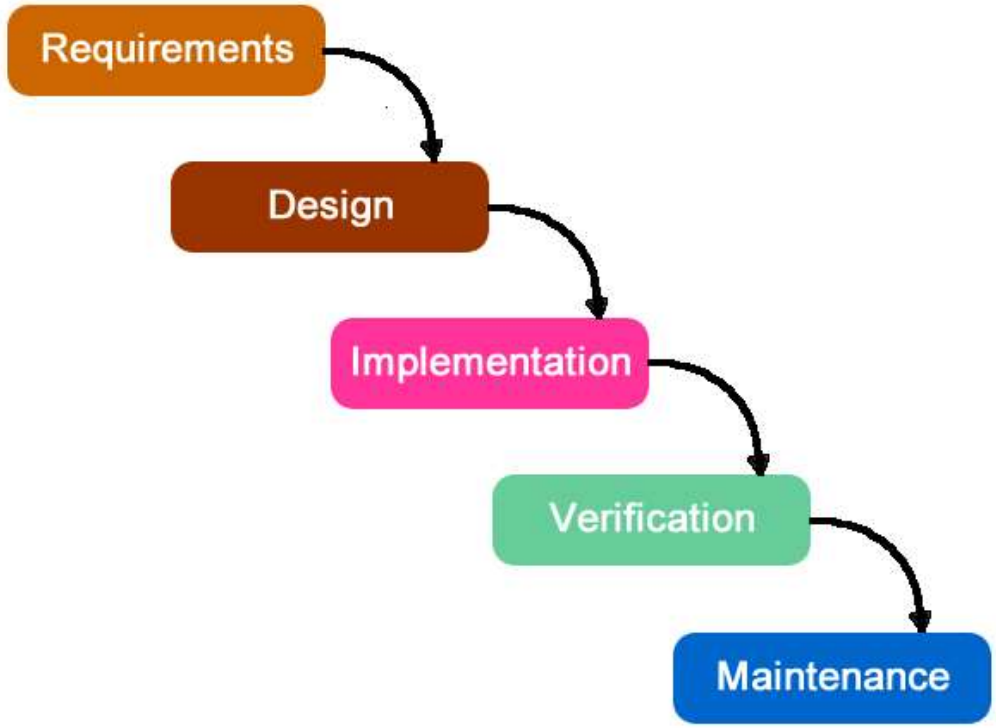
- Process of checking that the result is OK
  - **Unitary**: only one thing by requirement
  - **Complete**: no missing information
  - **Consistent**: no contradiction among requirements
  - **Unambiguous**: objective facts, comprehensible, …
  - **Prioritized**: level of importance is given
  - **Traceable**: source/reason/links are documented
  - **Verifiable**: can be checked at the end
  - …


- When requirement engineering?

# Requirements influence all the software activities

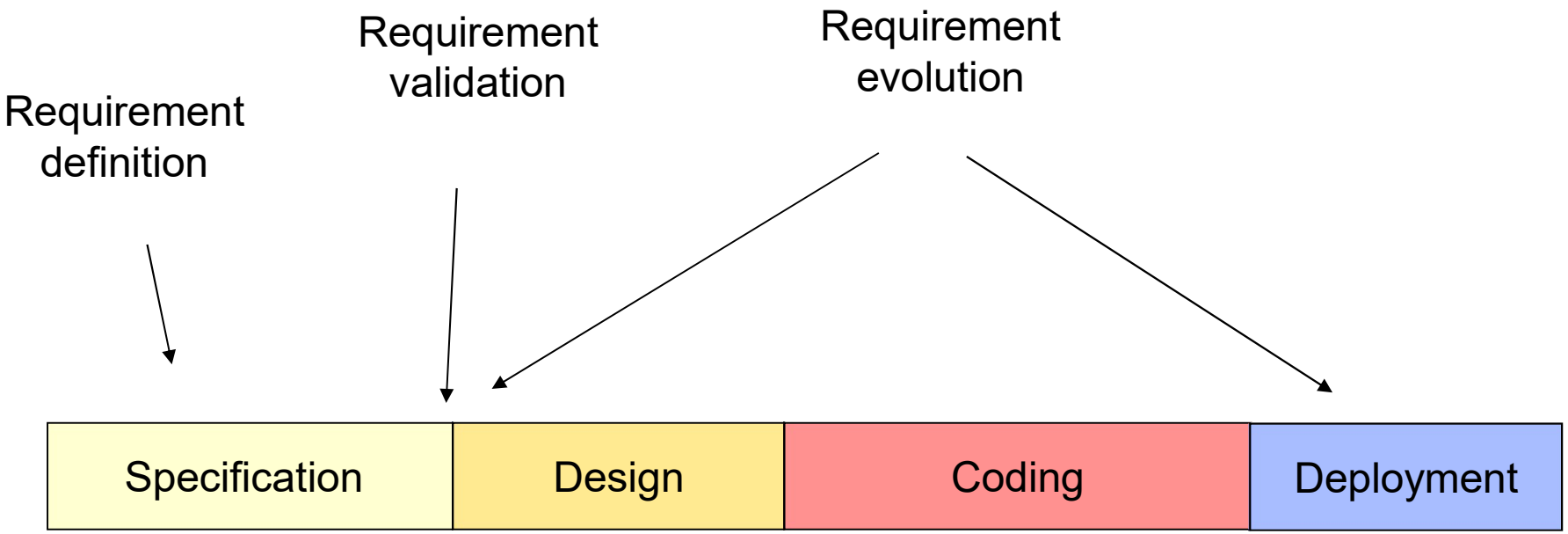
WHEN?

- Design
  - Architecture
  - Detailed design
- Implementation
- Validation
- Acceptance, ...



# Requirements and life cycle

WHEN?




**Requirement engineering**

# Requirement evolution and traceability


- User requirement may **evolve** (will)
- Should be **taken into account**
  - Possible if the initial elicitation work, analysis and validation has been carefully carried out
- **Impacts** should be evaluated
  - Possible only if traceability mechanisms
  - Tool can help to automate the links among the requirements

# Schedule

- What / Who / Why … requirement engineering
- Serious game: about requirement elicitation
- Serious game: try to do it
- More about Requirement specification

# About requirement elicitation method

- Make small groups (3-6 persons)
- Study an elicitation method
- Imagine a scenario of illustrate how this elicitation method is used
- Draw a strip cartoon to « implement » the scenario
- NO MORE THAN 1 HOUR !!!