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?

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?

- What are requirements?
- What is requirement engineering?

WHAT?

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

I CAN'T START THE PROJECT BECAUSE THE USER WON'T GIVE ME HIS REQUIREMENTS.
START MAKING SOMETHING ANYWAY. OTHERWISE WE'LL LOOK UNHELPFUL.
SO, OUR PLAN IS TO CLEVERLY HIDE OUR COMPETENCE.
YOU THINK TOO MUCH.
www.dilbert.com scottadams@aol.com
3/22/03 © 2003 United Feature Syndicate, Inc.

WHAT?

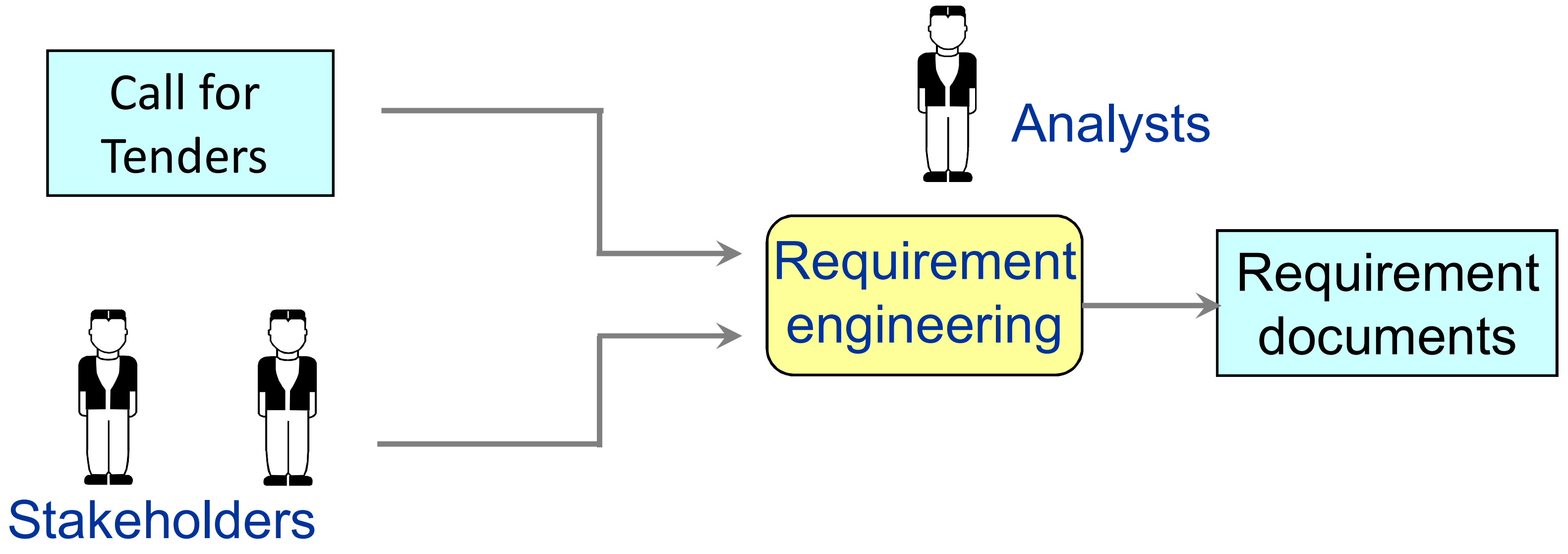
Requirements are **statements** of

- **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

WHAT?

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 the customer, the supplier, or both in a **Software Requirements Specification** document (SRS)

WHY?

- Why requirement engineering?

WHY?

Requirement Engineering

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

WHY?

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

I want something to get me across town in the shortest time

WHY?

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?

How is organized Requirement engineering?

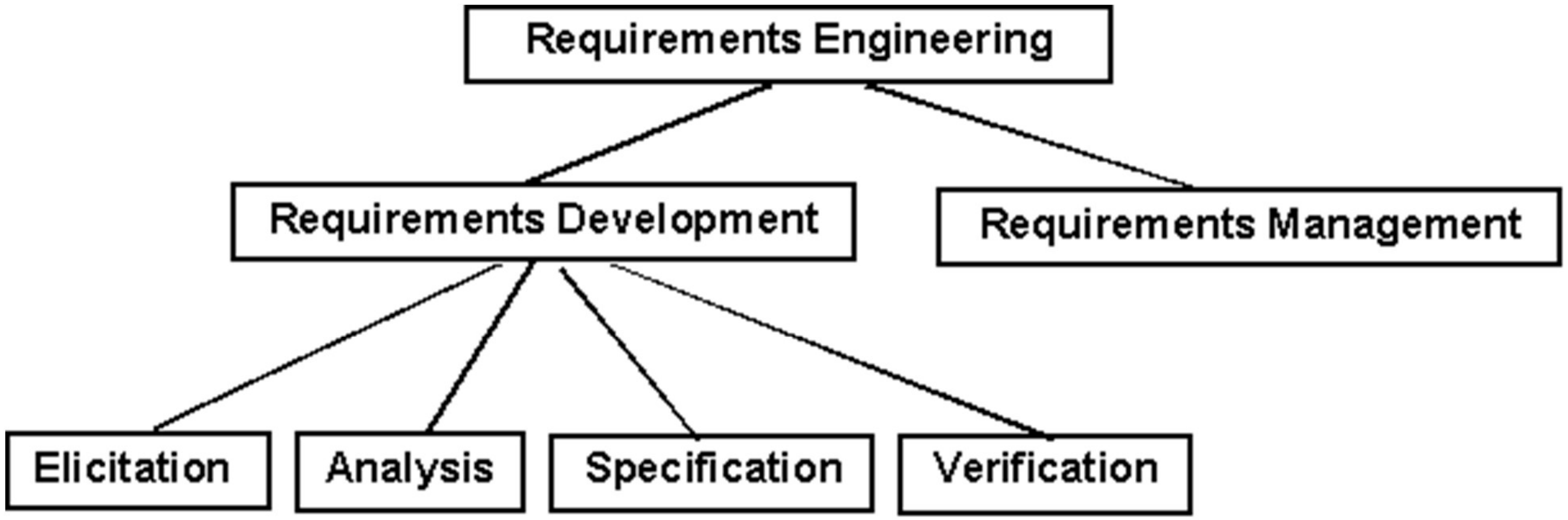


Figure 2. Subdisciplines of requirements engineering.

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

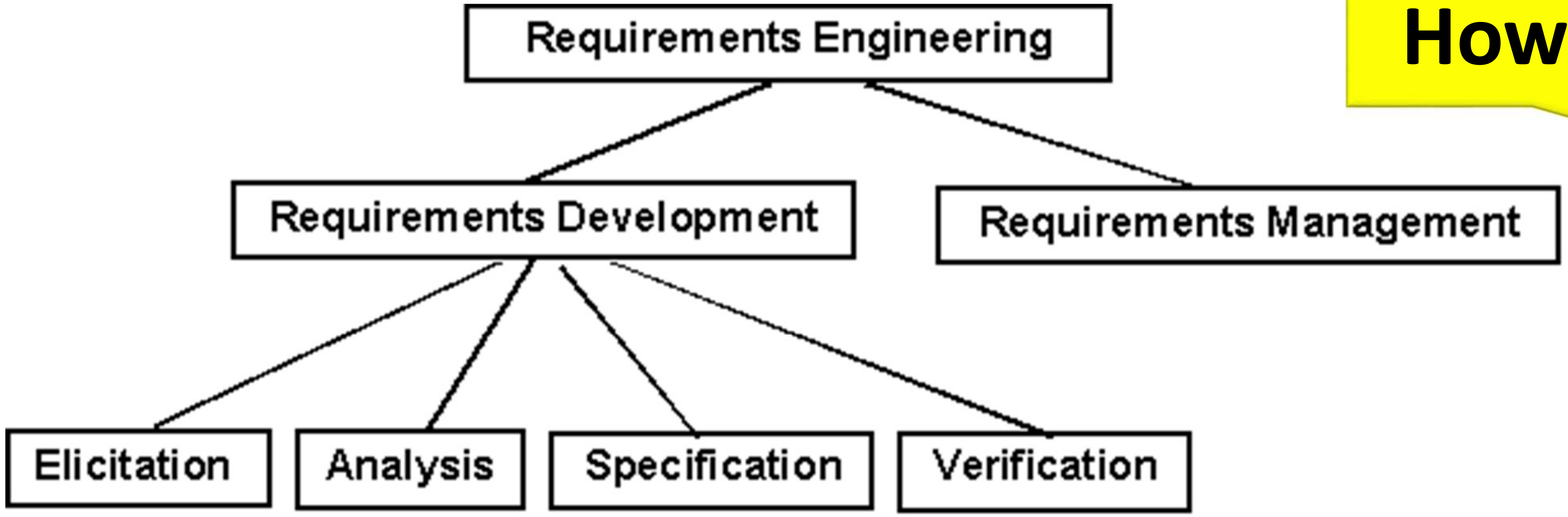
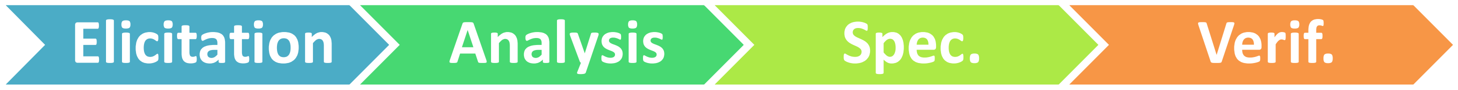


Figure 2. Subdisciplines of requirements engineering.

How?

How is organized Requirement Engineering?

REQUIREMENT DEVELOPMENT



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
  - Selecting techniques, approaches and tools 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)

Elicitation → Analysis → Spec. → Verif.

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

- Problem, 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

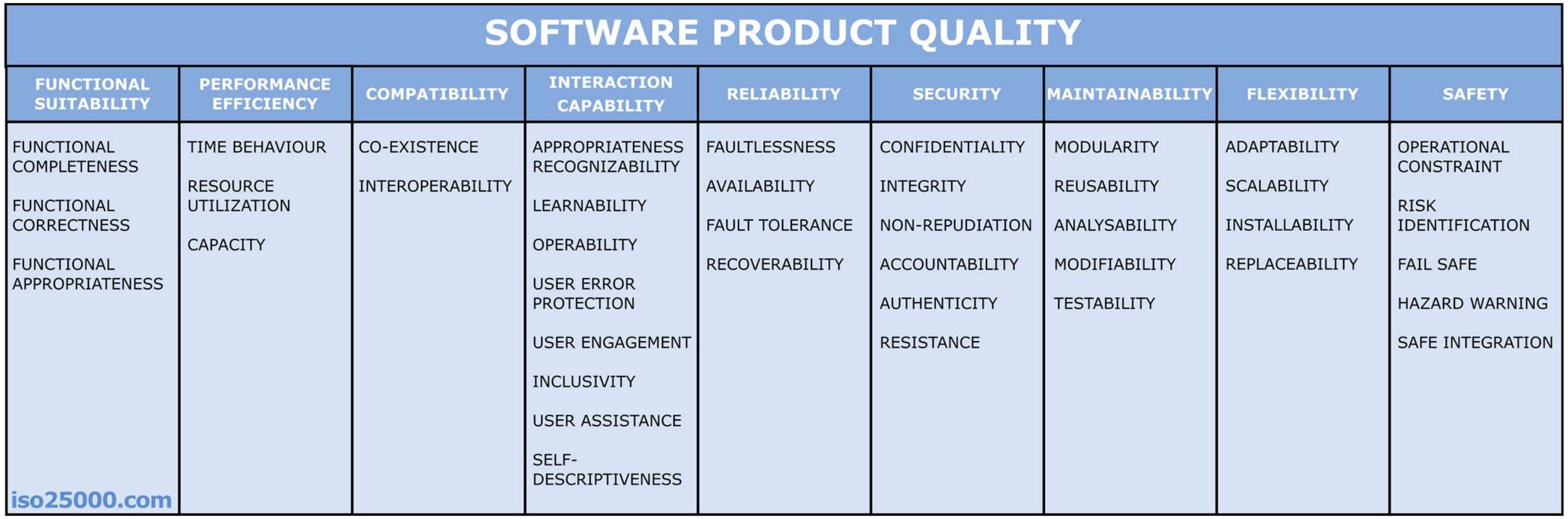
Requirement specification

- 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

Functional vs non-functional ISO/IEC 25010



SOFTWARE PRODUCT QUALITY								
FUNCTIONAL SUITABILITY	PERFORMANCE EFFICIENCY	COMPATIBILITY	INTERACTION CAPABILITY	RELIABILITY	SECURITY	MAINTAINABILITY	FLEXIBILITY	SAFETY
FUNCTIONAL COMPLETENESS	TIME BEHAVIOUR	CO-EXISTENCE	APPROPRIATENESS RECOGNIZABILITY	FAULTLESSNESS	CONFIDENTIALITY	MODULARITY	ADAPTABILITY	OPERATIONAL CONSTRAINT
FUNCTIONAL CORRECTNESS	RESOURCE UTILIZATION	INTEROPERABILITY	LEARNABILITY	AVAILABILITY	INTEGRITY	REUSABILITY	SCALABILITY	RISK IDENTIFICATION
FUNCTIONAL APPROPRIATENESS	CAPACITY		OPERABILITY	FAULT TOLERANCE	NON-REPUDIATION	ANALYSABILITY	INSTALLABILITY	FAIL SAFE
			USER ERROR PROTECTION	RECOVERABILITY	ACCOUNTABILITY	MODIFIABILITY	REPLACEABILITY	HAZARD WARNING
			USER ENGAGEMENT		AUTHENTICITY	TESTABILITY		SAFE INTEGRATION
			INCLUSIVITY		RESISTANCE			
			USER ASSISTANCE					
			SELF-DESCRIPTIVENESS					

iso25000.com

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
  - …

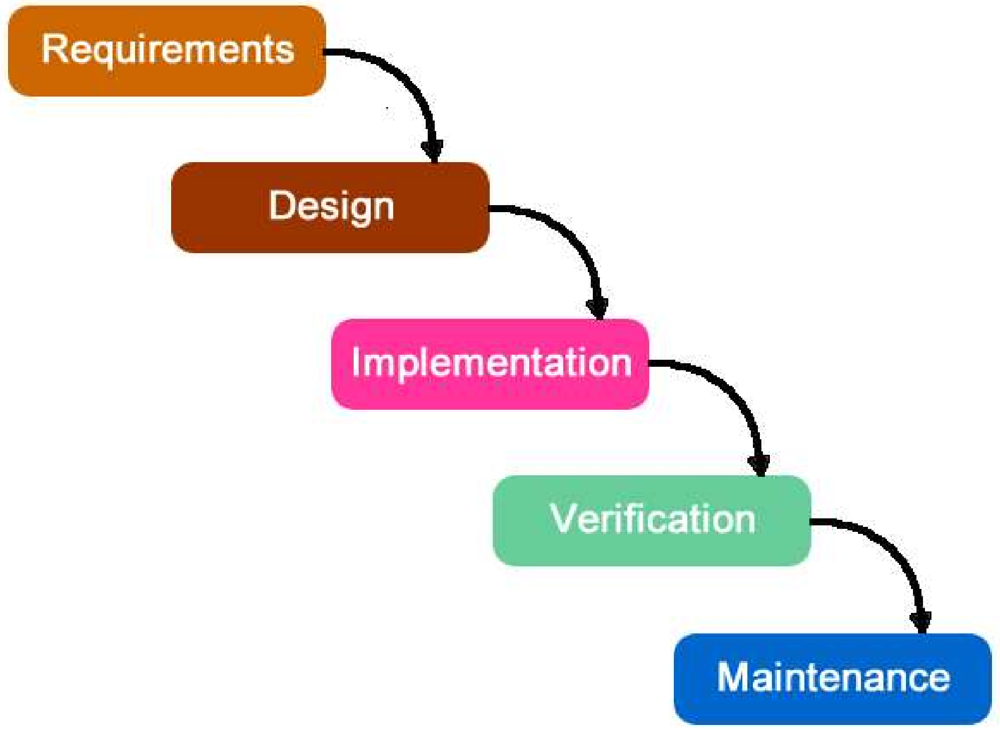
WHEN?

- When requirement engineering?

WHEN?

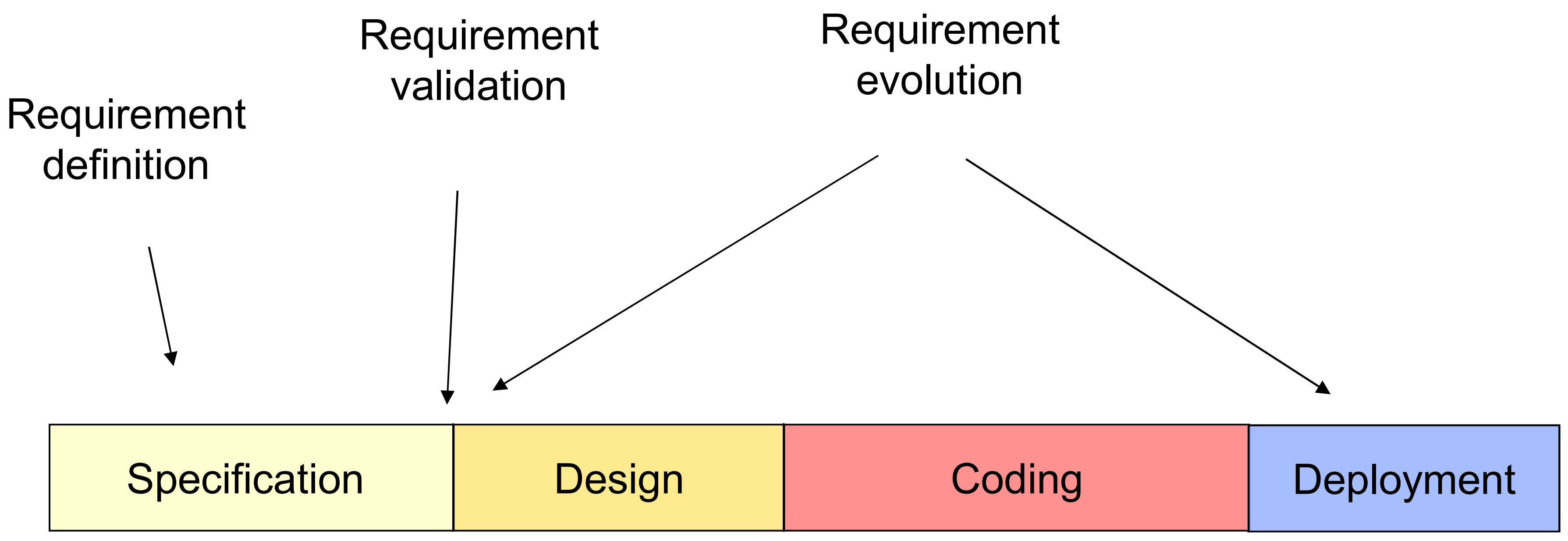
Requirements influence all the software activities

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



WHEN?

Requirements and life cycle



Requirement engineering

WHEN?

Requirement evolution and traceability

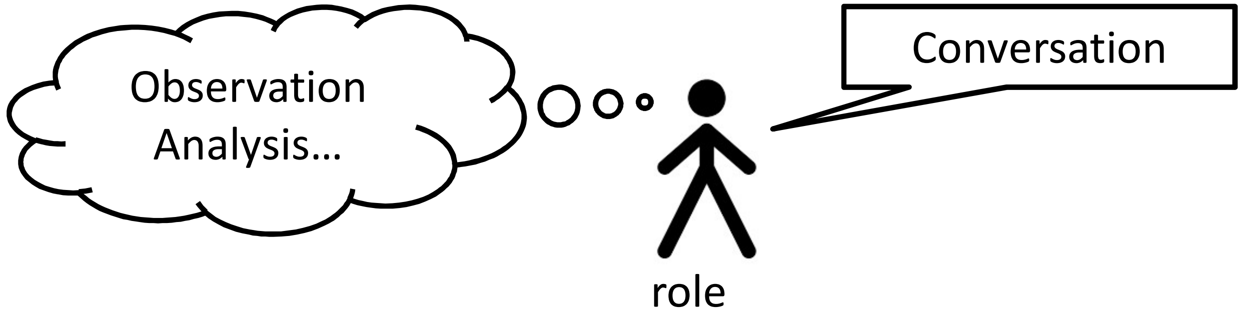
- User requirement may **evolve** (will)
- Should be **taken into account**
  - Possible if the initial elicitation work, analysis and validation has been carefully carried out
- **Impacts** should be evaluated
  - Possible only if traceability mechanisms
  - Tool can help to automate the links among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

- Phase 1 :
  - Make small groups (2-3 persons)
  - Study an elicitation method
  - Imagine a scenario of illustrate
    how this elicitation method is used
  - Draw a strip cartoon to « implement » the scenario
  - Keep simple : at most one page
  - Put your names but do not name the method on the strip



About requirement elicitation method

- Phase 2
  - Exchange your strip with another group
  - Guess the elicitation method
  - Make some constructive comments on the back if appropriate