



---

# The requirements engineering process

Slides used in the video available here

[https://polimi365-my.sharepoint.com/:v:/g/personal/10143828\\_polimi\\_it/Eb4Mp2L\\_i6tGqX6\\_TtFepmwB7e9nytlwrBquqqmVI7sJ8A?e=Gx0gMn](https://polimi365-my.sharepoint.com/:v:/g/personal/10143828_polimi_it/Eb4Mp2L_i6tGqX6_TtFepmwB7e9nytlwrBquqqmVI7sJ8A?e=Gx0gMn)

---

# Issues concerning RE

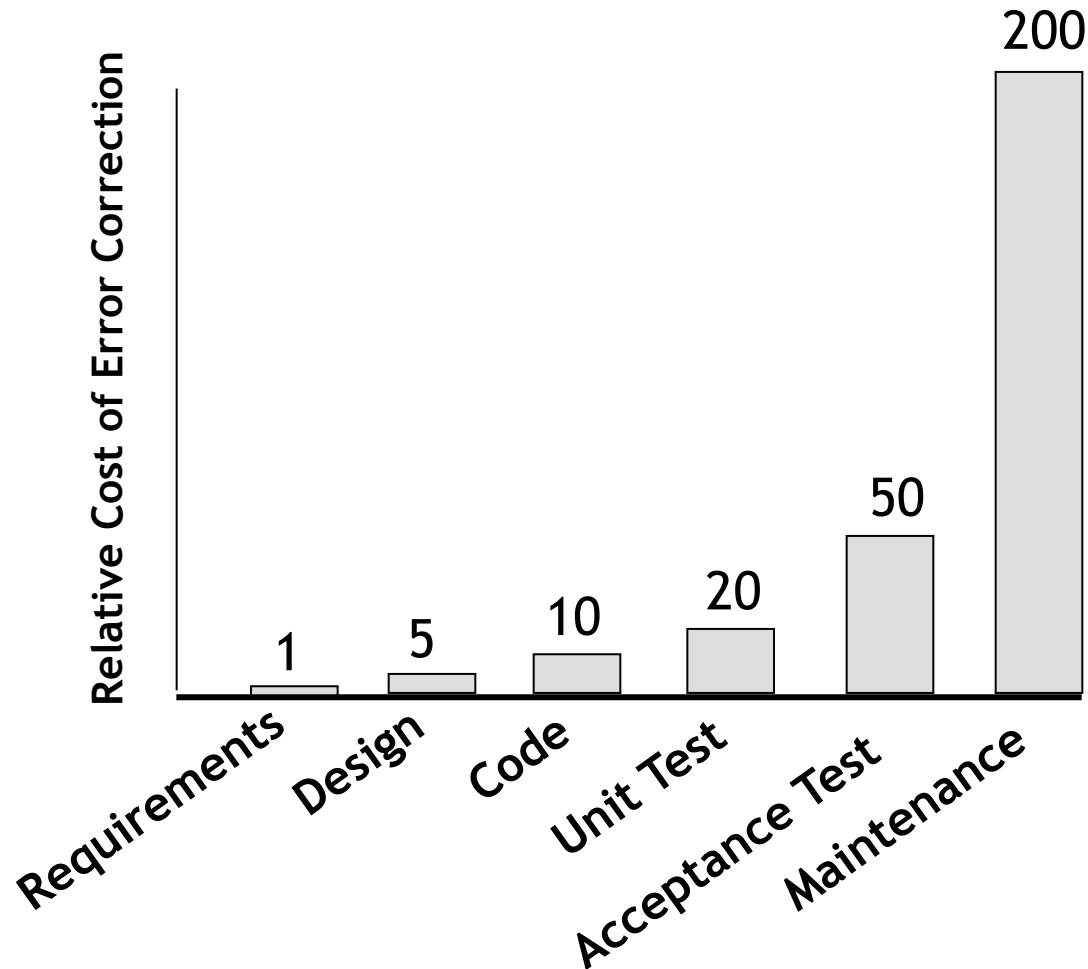
---



- Poor requirements are ubiquitous ...
    - ▶ “The system should guarantee that the reserved taxi picks the user up”: is this reasonable?
    - ▶ “[...] requirements need to be *engineered* and have continuing review & revision” (Bell & Thayer, empirical study, 1976)
  - RE is hard & critical ....
    - ▶ “[...] hardest, most important function of SE is the iterative **extraction & refinement** of requirements” (F. Brooks, 1987)
    - ▶ Does the list in the previous slide capture all needs of the customer?
  - Prohibitive cost of late correction ...
    - ▶ “up to 200 x cost of early correction” (Boehm, 1981)
-



# Cost of late correction (Boehm, 1981)



The cost of correcting an error depends on the number of subsequent decisions that are based on it

Errors made in understanding requirements have the potential for greatest cost, because many other design decisions depend on them



# Impact of RE

- Survey of US software projects by Standish Group

	1994	1998
Successful	16%	26%
Challenged	53%	46%
Cancelled	31%	28%

	Successful	Challenged	Cancelled
1.	User involvement	Lack of user input	Incomplete requirements
2.	Executive management support	Incomplete requirements	Lack of user input
3.	Clear statement of requirements	Changing requirements	Lack of resources

# What makes RE so complex ? (1)

---



- Broad scope
  - ▶ **Composite** systems: human organizations + physical devices + software components
  - ▶ **More than one system**: system-as-is, alternative proposals for system-to-be, system evolutions, product family
  - ▶ **Multiple abstraction levels**: high-level goals, operational details

# What makes RE so complex ? (2)

---



- **Multiple concerns**

- ▶ functional, quality, development
- ▶ hard and soft concerns

→ *conflicts*

- **Multiple stakeholders with different background**

- ▶ clients, users, domain experts, developers, ...

→ *conflicts*

# What do requirements engineers do? (1)

---



- Eliciting Information

- ▶ Project objectives, context and scope
- ▶ Domain knowledge and requirements

- Modelling and analysis

- ▶ Goals, objects, use cases, scenarios, ...

- Communicating requirements

- ▶ Analysis feedback, RASD document, system prototypes, ...
-

# What do requirements engineers do? (2)

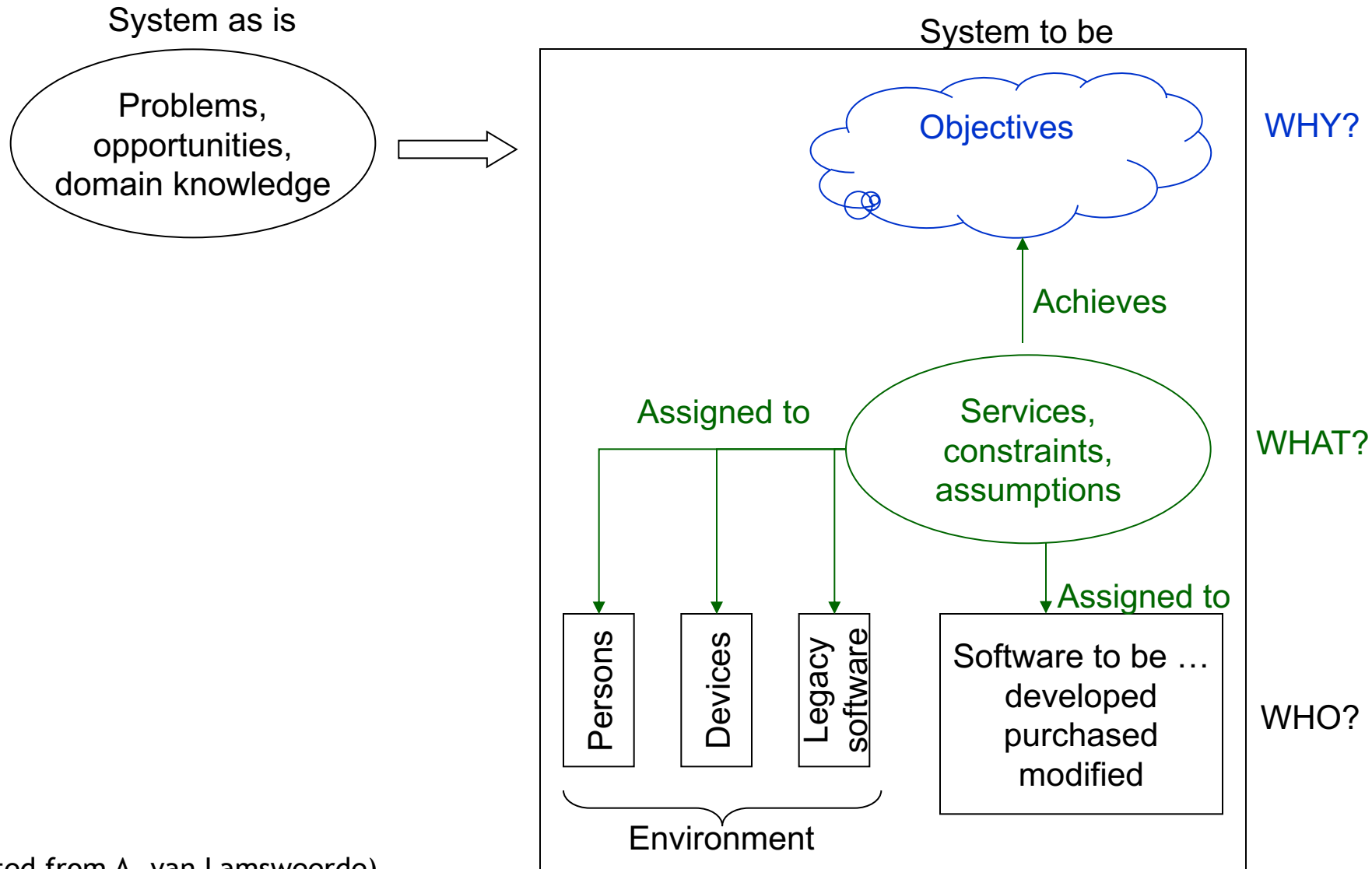
---



- Negotiating and agreeing Requirements
    - ▶ Handling conflicts and risks
    - ▶ Helping in requirement selection and prioritization
  - Managing and evolving Requirements
    - ▶ Managing requirements during development: backward and forward traceability
    - ▶ Managing requirements changes and their impacts
-



# The dimensions of RE



(adapted from A. van Lamsweerde)

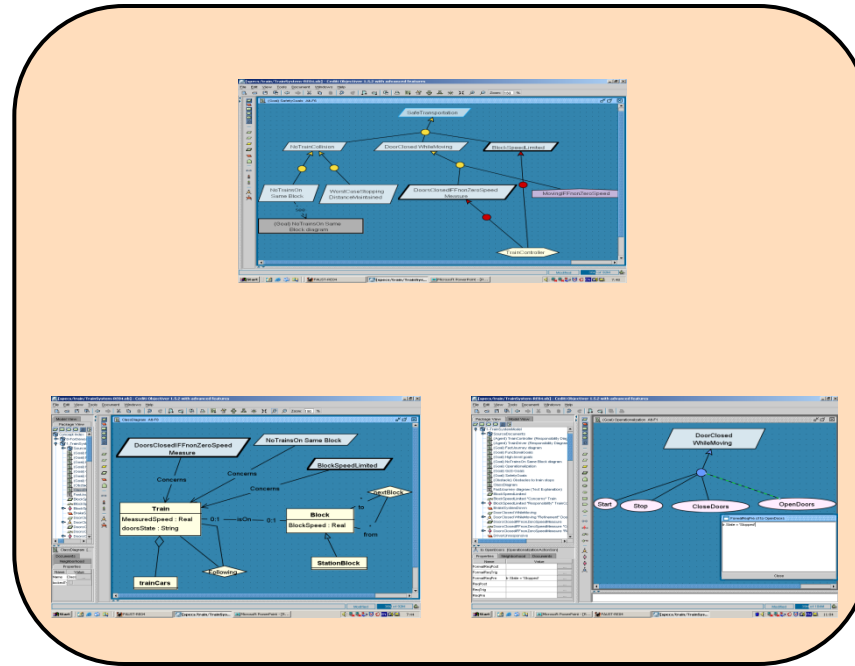
# Role of requirements models



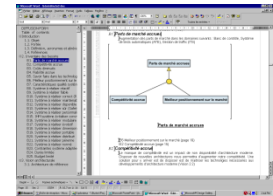
elicitation  
& modelling



## Requirements Models



generation of  
RE deliverables



requirements  
document



analysis  
& validation