

# Gathering Requirements

# Gathering Requirements

Average length of sentences in an essay

Which question is most informative?

- What OS(es) should it run on?
- How will the user specify input?
- How many lines of code should it take?

# Average Sentence Length

- Vanilla Java 1.6
- Compile with javac, no options
- Executable from command line
- User specifies file path
- User specifies delimiters with –d
- User specifies word length limit w/ -l
- Output average sentence length

# **SOFTWARE ENGINEERING**

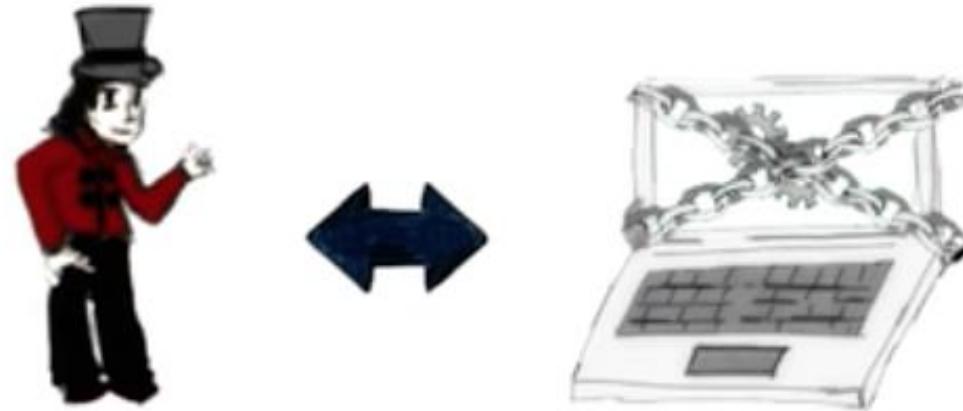
## **REQUIREMENTS ENGINEERING**

# JANE CLELAND-HUANG

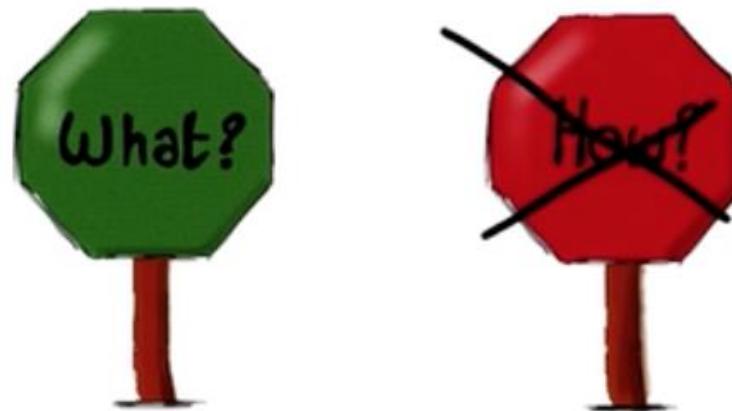


# Requirements Engineering (RE)

= =

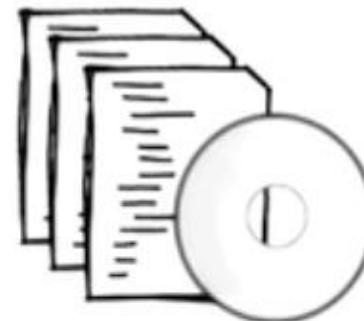


- => Software Requirements Specification (SRS)



# Software Intensive Systems

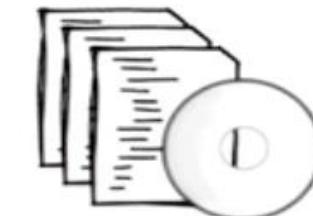
software



software intensive system = Software + hardware + Context



=



+



+



# Software Quality



Software



hardware



Purpose

runs on some

that is related to human activities

$$\text{Quality} = f(\text{Software}, \text{Hardware}, \text{Purpose})$$

FITNESS  
FOR PURPOSE



Requirements engineering is mostly about identifying the purpose

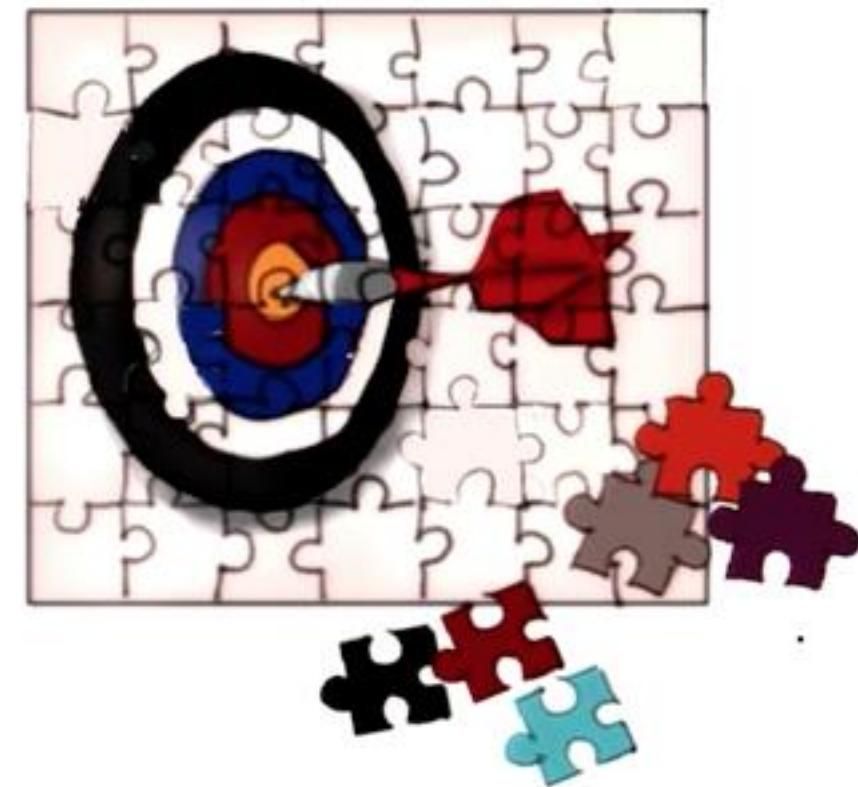
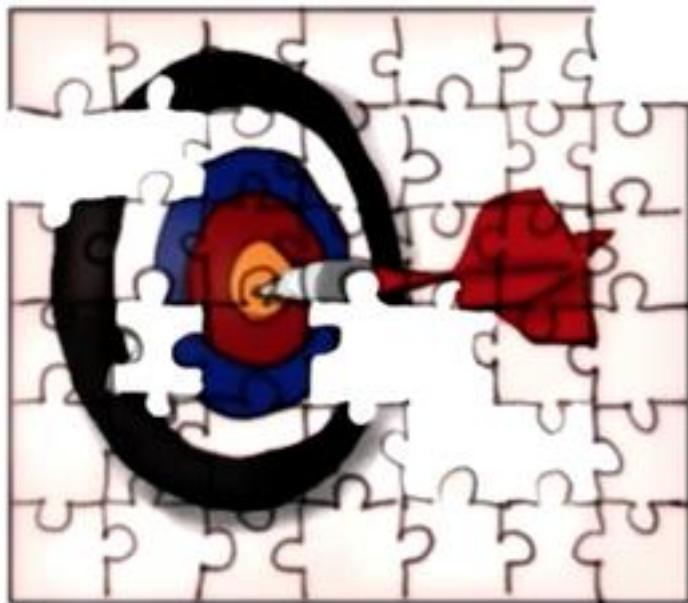
# Identifying Purpose = Defining Requirements



Extremely hard task!

- Sheer complexity of the purpose / requirements
- Often, people don't know what they want until you show it to them
- Changing requirements
- Multiple stakeholders with conflicting requirements

# Completeness and Pertinence





Consider an information system for a gym. In the list below, mark all the requirements that you believe are pertinent

- Members of the gym shall be able to access their training programs
- The system shall be able to read member cards
- The system shall be able to store members' commute time
- Personal trainers shall be able to add clients
- The list of members shall be stored as a linked list



Consider an information system for a gym. In the list below, mark all the requirements that you believe are pertinent

- Members of the gym shall be able to access their training programs
- The system shall be able to read member cards
- The system shall be able to store members' commute time
- Personal trainers shall be able to add clients
- The list of members shall be stored as a linked list



Consider an information system for a gym. In the list below, mark all the requirements that you believe are pertinent

- Members of the gym shall be able to access their training programs
- The system shall be able to read member cards
- The system shall be able to store members' commute time
- Personal trainers shall be able to add clients
- The list of members shall be stored as a linked list

Is the above list complete?

- Yes
- No



Consider an information system for a gym. In the list below, mark all the requirements that you believe are pertinent

- Members of the gym shall be able to access their training programs
- The system shall be able to read member cards
- The system shall be able to store members' commute time
- Personal trainers shall be able to add clients
- The list of members shall be stored as a linked list

Is the above list complete?

- Yes
- No



Why can irrelevant requirements be harmful?

- [ ] They may lead to missing functionality in the final product
- [ ] They can introduce inconsistency
- [ ] They can waste project resources
- [ ] They may introduce bugs in the software system



## Why can irrelevant requirements be harmful?

- They may lead to missing functionality in the final product
- They can introduce inconsistency
- They can waste project resources
- They may introduce bugs in the software system

# Best Practice?



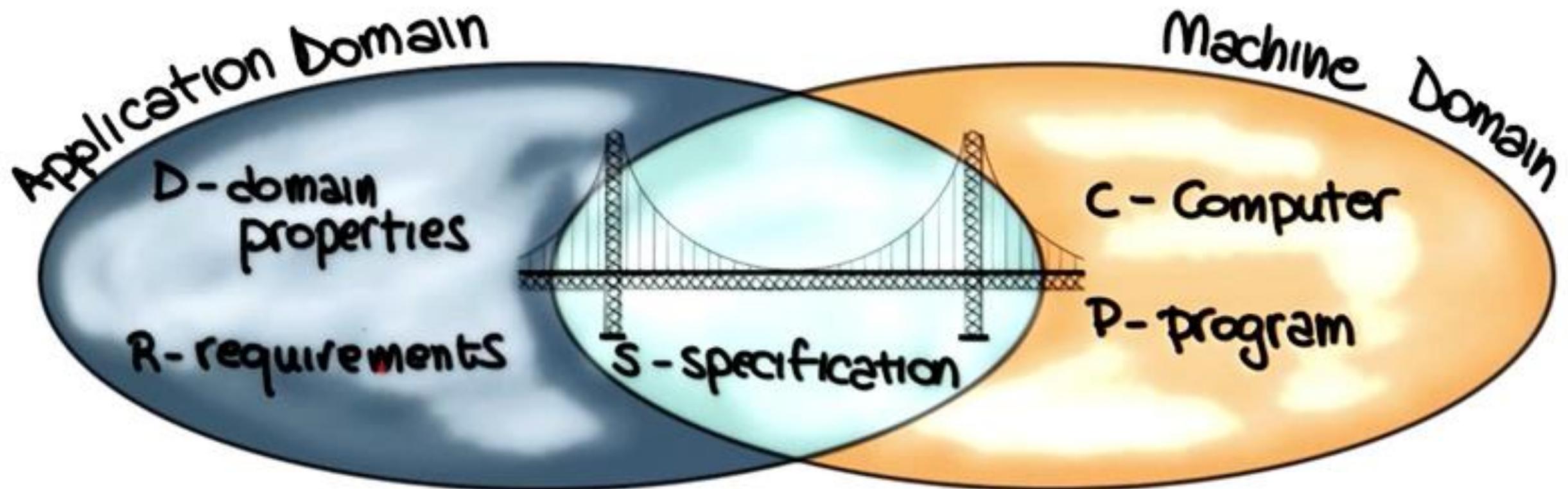
# Definition of Requirements Engineering

- Requirement Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the context in which it will be used.
- Hence, RE acts as the bridge between the real-world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies

# Definition of Requirements Engineering

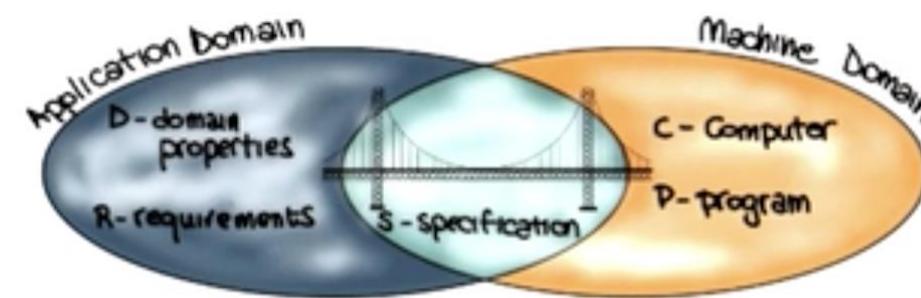
- Requirement Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the context in which it will be used.
- Hence, RE acts as the bridge between the real-world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies

# What are Requirements?





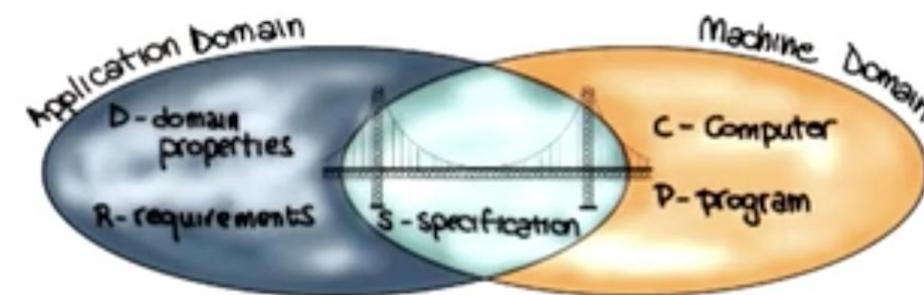
Referring to the figure that we just discussed, indicate, for each of the following items, whether they belong to the machine domain (1), application domain (2), or their intersection (3) (Enter the corresponding number - 1, 2, or 3 - in the entry next to the item)



- [ ] An algorithm sorts a list of books in alphabetical order by the first author's name
- [ ] A notification of the arrival of a message appears on a smart watch
- [ ] An employee wants to organize a meeting with a set of colleagues
- [ ] A user clicks a link on a web page



Referring to the figure that we just discussed, indicate, for each of the following items, whether they belong to the machine domain (1), application domain (2), or their intersection (3) (Enter the corresponding number - 1, 2, or 3 - in the entry next to the item)



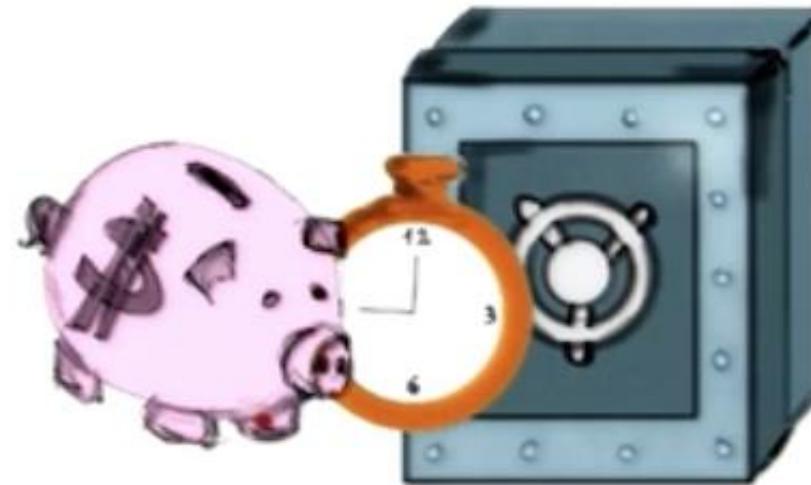
- [1] An algorithm sorts a list of books in alphabetical order by the first author's name
- [3] A notification of the arrival of a message appears on a smart watch
- [2] An employee wants to organize a meeting with a set of colleagues
- [3] A user clicks a link on a web page

# Functional and Non-Functional Requirements

Functional



Non-functional



# User and System Requirements



## User Requirements

- written for customers
- often in natural language, no technical details

## System requirements



- written for developers
- detailed functional and non-functional requirements
- clearly and more rigorously specified



## User Requirements Definition

The software must provide a means of representing and accessing external files created by other tools.



## System Requirements Definition

- 1.1 The user should be provided with facilities to define the type of external files
- 1.2 Each external file type may have an associated tool which may be applied to the file.
- 1.3 Each external file type may be represented as a specification from the user's display.
- 1.4 Facilities should be provided for the icon representing an external file type to be defined by the user.
- 1.5 When the user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of the external file to the file represented by the selected icon.



Which of the following requirements are non-functional requirements?

- [ ] The BowlingAlley program keeps track of the score during a game
- [ ] The WordCount program should be able to process large files
- [ ] The Login program for a website should be secure
- [ ] The VendingMachine program should take coins as an input from the user.



Which of the following requirements are non-functional requirements?

- [ ] The BowlingAlley program keeps track of the score during a game
- [X] The WordCount program should be able to process large files
- [X] The Login program for a website should be secure
- [ ] The VendingMachine program should take coins as an input from the user.





# WHERE DO REQUIREMENTS COME FROM?

Stakeholders



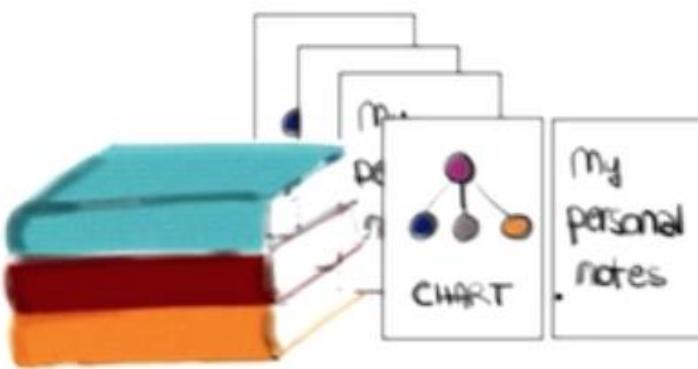
...

Application domain



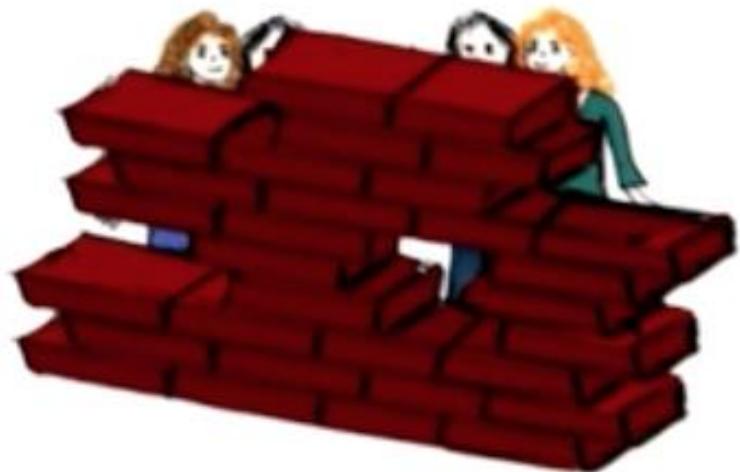
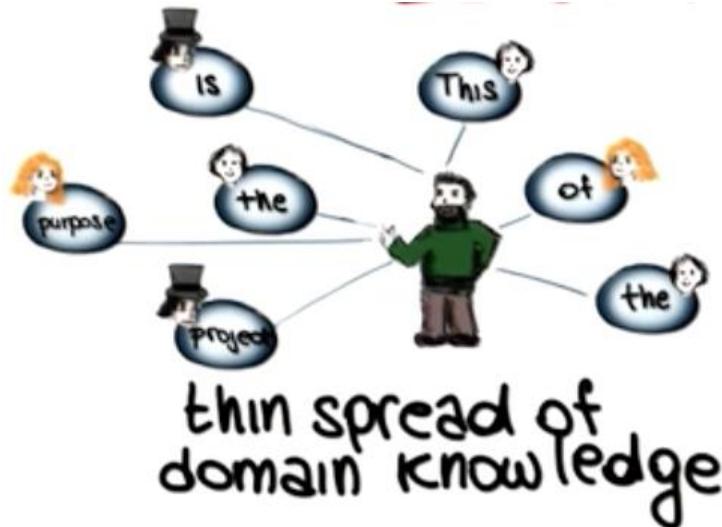
...

Documentation

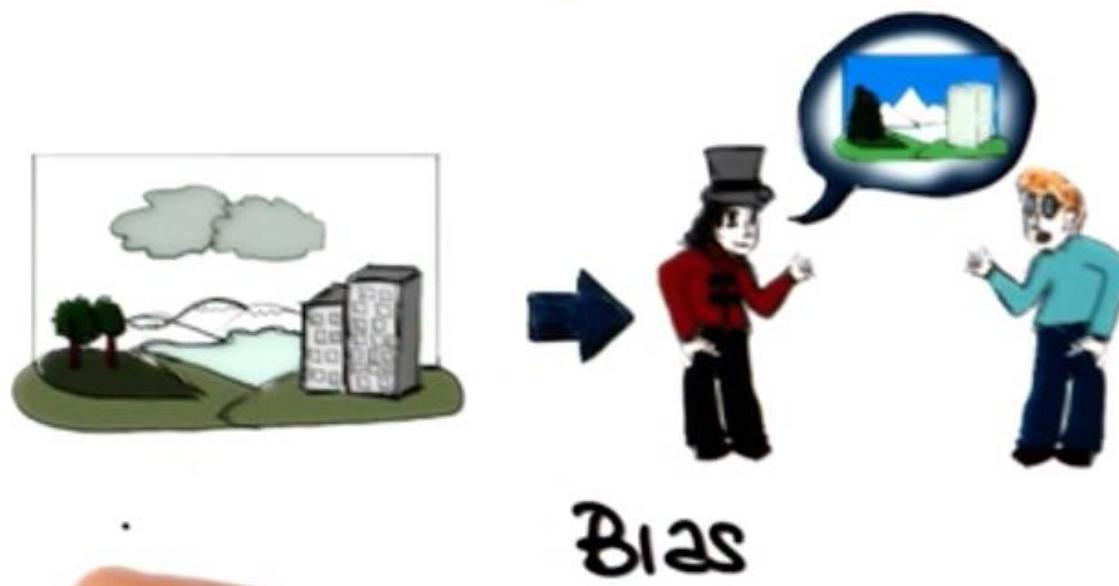
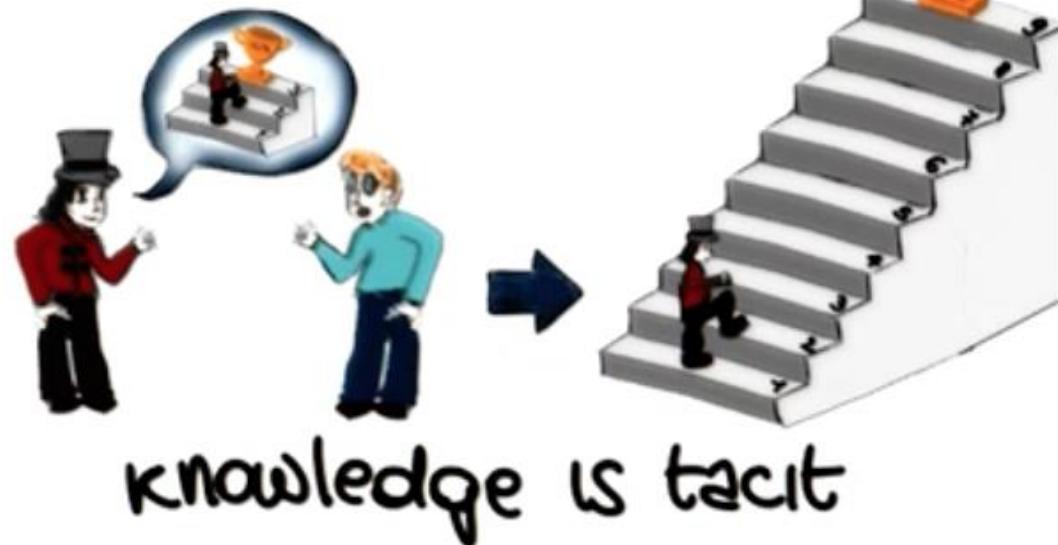


...

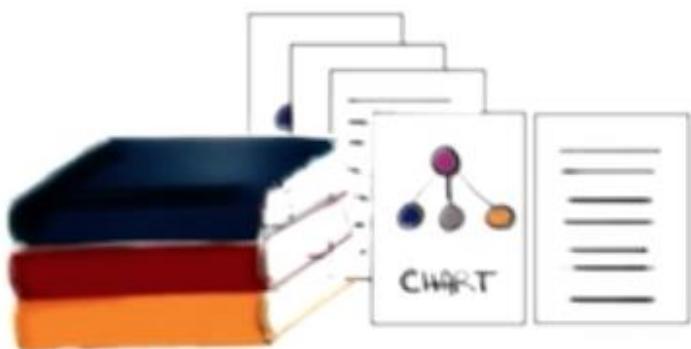
# ELICITATION PROBLEMS



Limited observability



# Traditional Techniques



Background reading



Interviews



surveys



Hard data and Samples



Meetings

# Traditional Techniques

Collaborative techniques



Social approaches



Cognitive techniques



# MODELING REQUIREMENTS



## Modeling enterprises

- goals and objectives
- organizational structure
- task and dependencies
- agents, roles, intentionality



Organization modeling  
I<sup>n</sup>, soft system modeling ...  
Goal modeling  
KAOS, CREWS ...

## Modeling information and behaviors

- information structure
- behavioral view
  - scenarios and use cases
  - state machine models
  - Sequence diagrams
  - information flow
- time / sequencing requirements



Information modeling  
E-R, class diagram ...  
Structure analysis  
Struc. Analysis and design tech. (SAD)  
Jackson software development ...  
Object oriented analysis  
UML  
Formal methods  
Alloy, Petri Net, Z ...

## Modeling system qualities (NFRs)

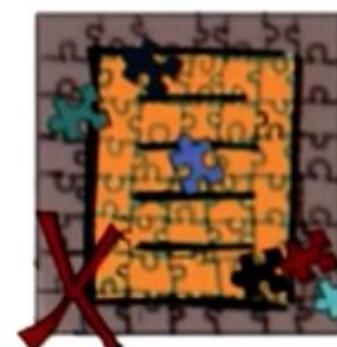
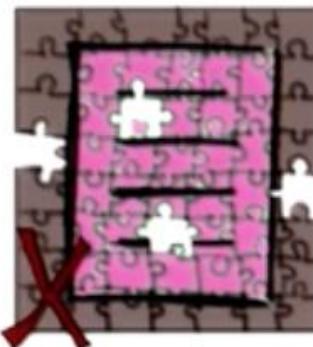
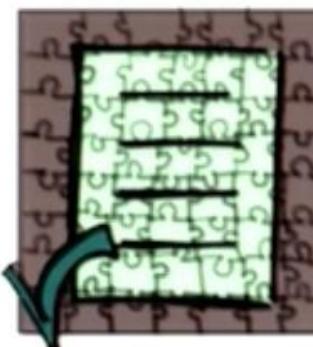


Quality tradeoffs  
Win-win, NFR, Analytic Hierarchy process (AHP) ...  
Specific NFRs  
Timed Petri net (performance)  
Task models (usability) ...

# ANALYZING REQUIREMENTS

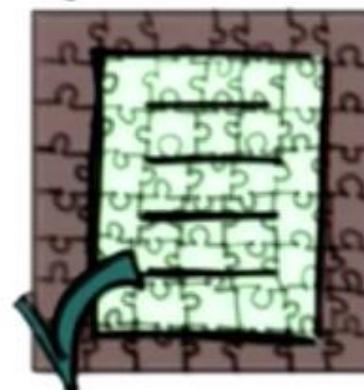


Verification



...

Validation



...

Risk analysis



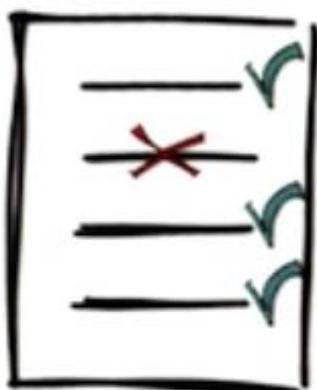
# REQUIREMENTS PRIORITIZATION



Limited resources



⇒ inability to satisfy all the requirements



⇒ need to prioritize them

mandatory  
nice to have  
superfluous





Imagine that you have collected the following set of five requirements for an ATM system but only have resources to satisfy two, possibly three of those.

Suitably prioritize the requirements by marking them as mandatory (M), nice to have (N), or superfluous (S).

- [ ] The system shall check the PIN of the ATM card before allowing the customer to perform an operation
- [ ] The system shall perform an additional biometric verification of the customer's identity before it allows the customer to perform an operation
- [ ] The system shall allow customers to withdraw cash using an ATM card
- [ ] The system shall allow customers to deposit money using an ATM card
- [ ] The system shall allow customers to change the PIN of an ATM card



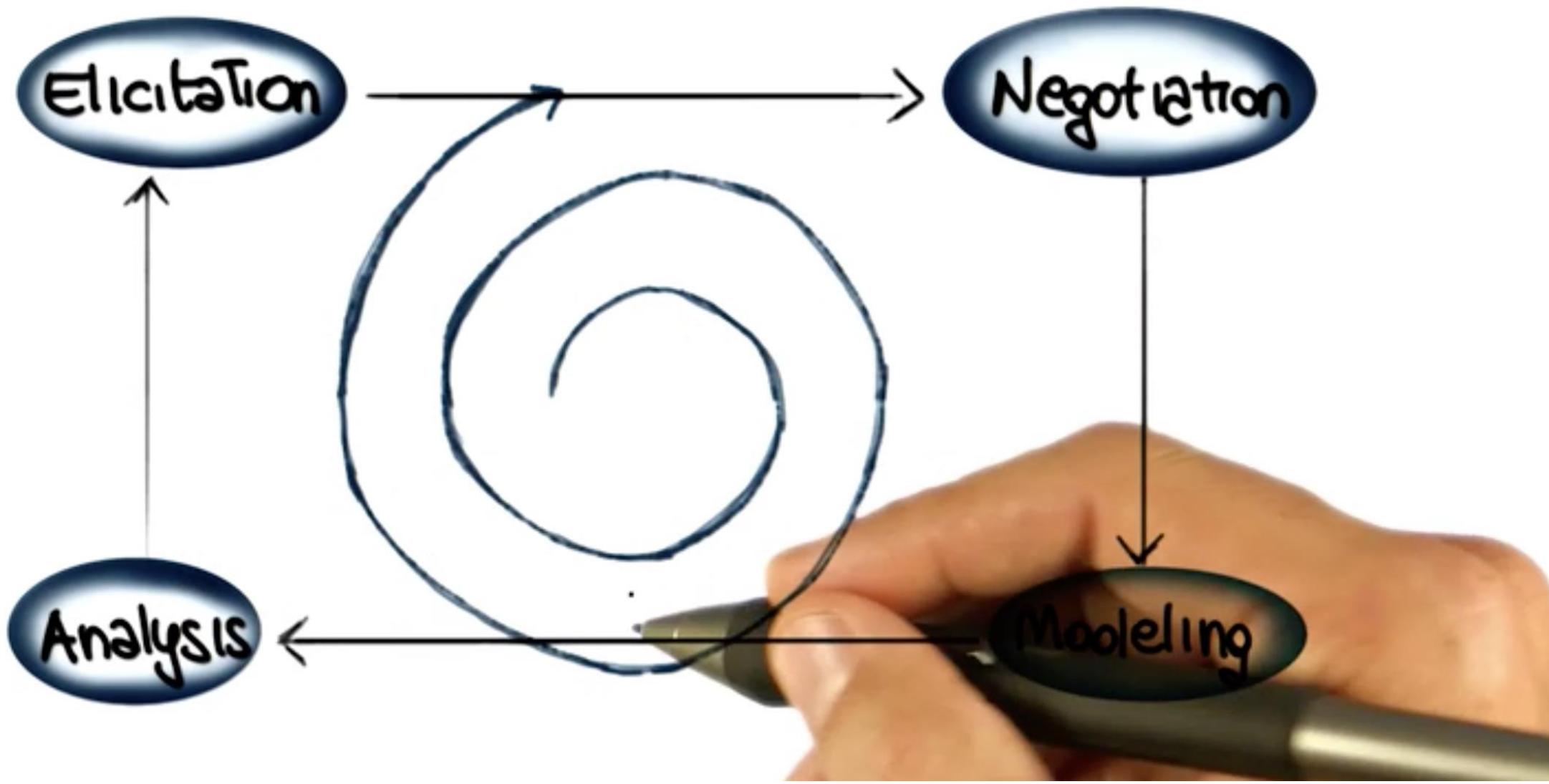
Imagine that you have collected the following set of five requirements for an ATM system but only have resources to satisfy two, possibly three of those.

Suitably prioritize the requirements by marking them as mandatory (M), nice to have (N), or superfluous (S).

- [M] The system shall check the PIN of the ATM card before allowing the customer to perform an operation
- [S] The system shall perform an additional biometric verification of the customer's identity before it allows the customer to perform an operation
- [M] The system shall allow customers to withdraw cash using an ATM card
- [N] The system shall allow customers to deposit money using an ATM card
- [S] The system shall allow customers to change the PIN of an ATM card

# REQUIREMENTS ENGINEERING PROCESS

ITERATIVE



# Software Requirements Specification(SRS)

## Properties

- simple (not compound)
- testable
- organized
- numbered (for traceability)

different SRS's

