CS3300 Introduction to Software Engineering

Lecture 07: Requirements Engineering

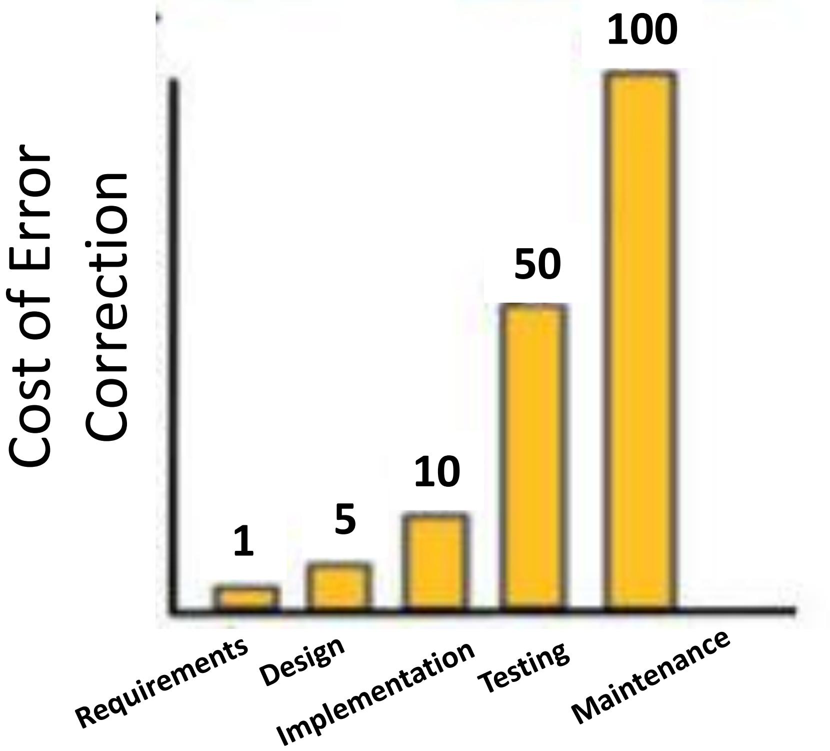
Nimisha Roy ▸ nroy9@gatech.edu

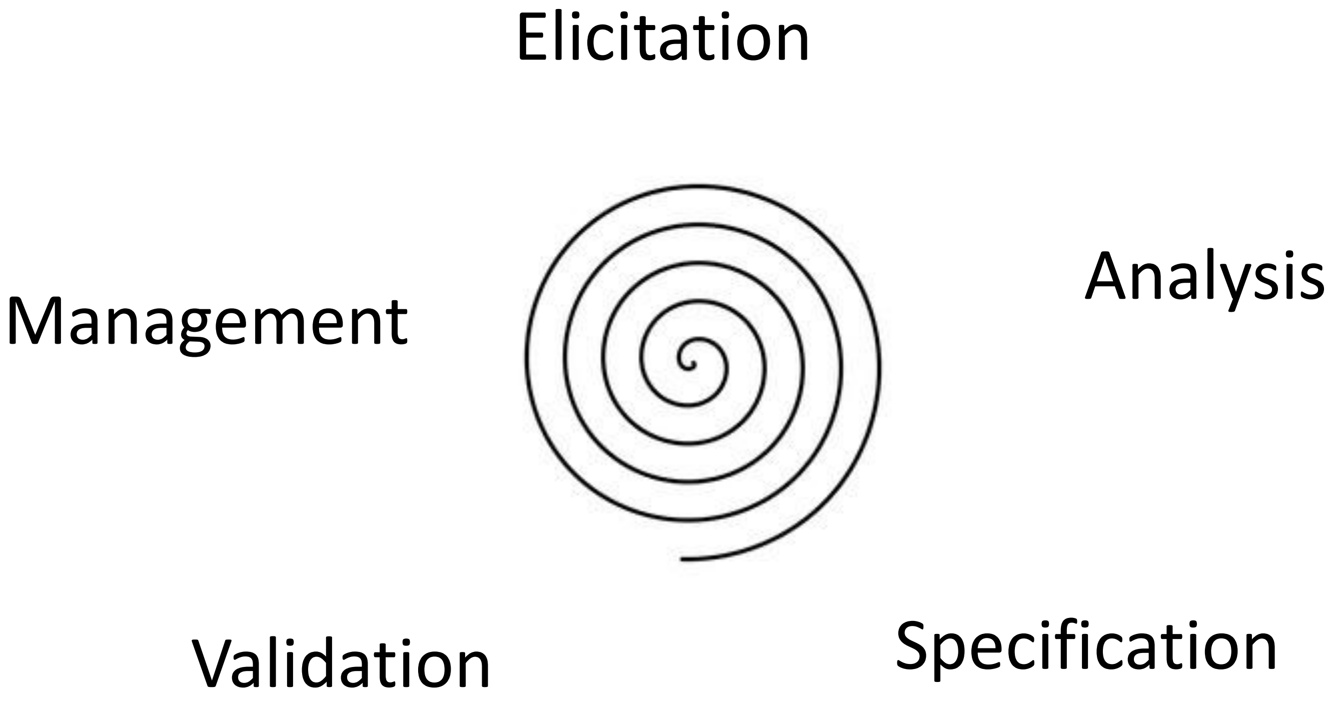
Slides adapted from Alessandro Orso

Requirements Engineering (RE)

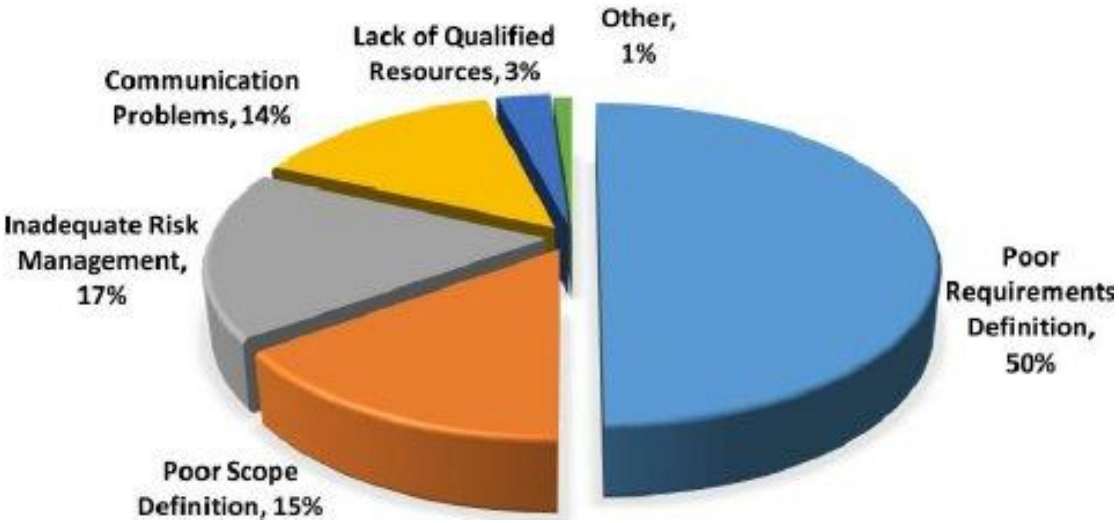
RE is the process of establishing the needs of stakeholders that are to be solved by software

Cost of Late Correction



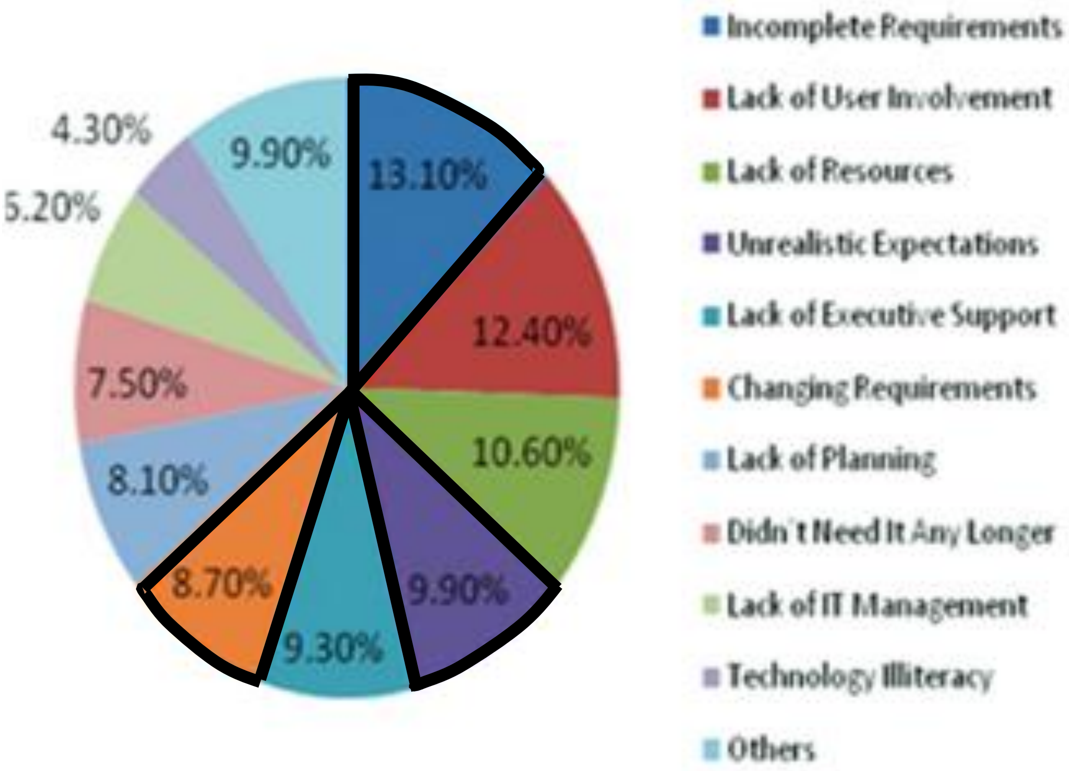


Software Failures due to RE



Role of Requirements in Software Project Failures (Agile)

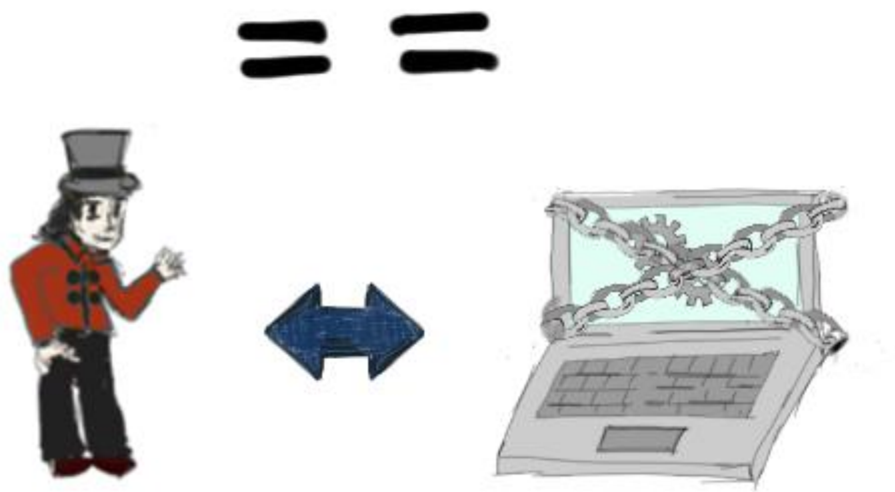
Source: Abdou, T., Kamthan, P., & Shahmir, N. (2014). User Stories for Agile Business: INVEST, Carefully!. Social Media and Publicity, 141.



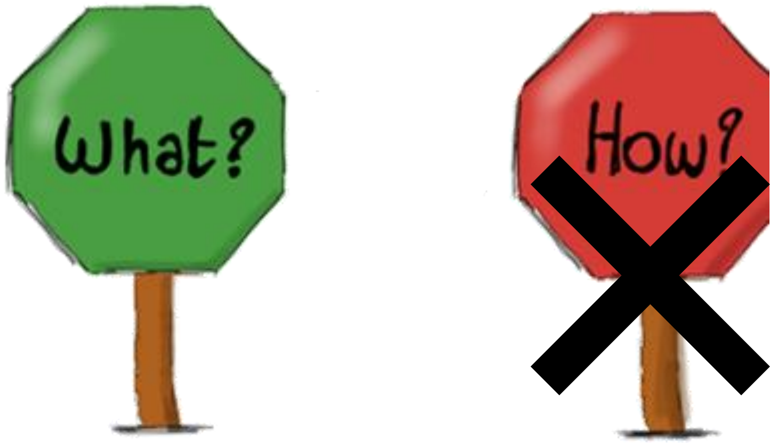
Role of Requirements in Software Project Failures.

Source: Hussain, A., Mkpojiogu, E. O., & Kamal, F. M. (2016). The role of requirements in the success or failure of software projects. International Review of Management and Marketing, 6(7S), 306-311

Requirements Engineering (RE)

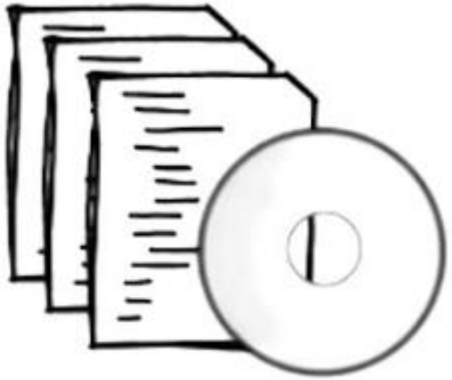


=> Software Requirements Specification (SRS)

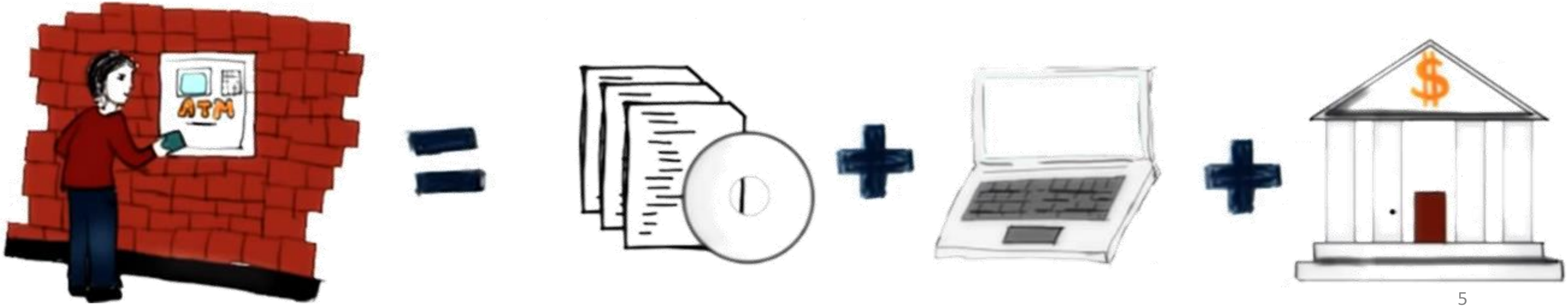


Software Intensive Systems

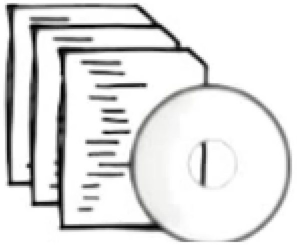
Software



Software Intensive System = Software + Hardware + Context



Software Quality



Software runs on some hardware and is developed for a purpose that is related to human activities

Quality ≠ f( )

Quality = f(,)

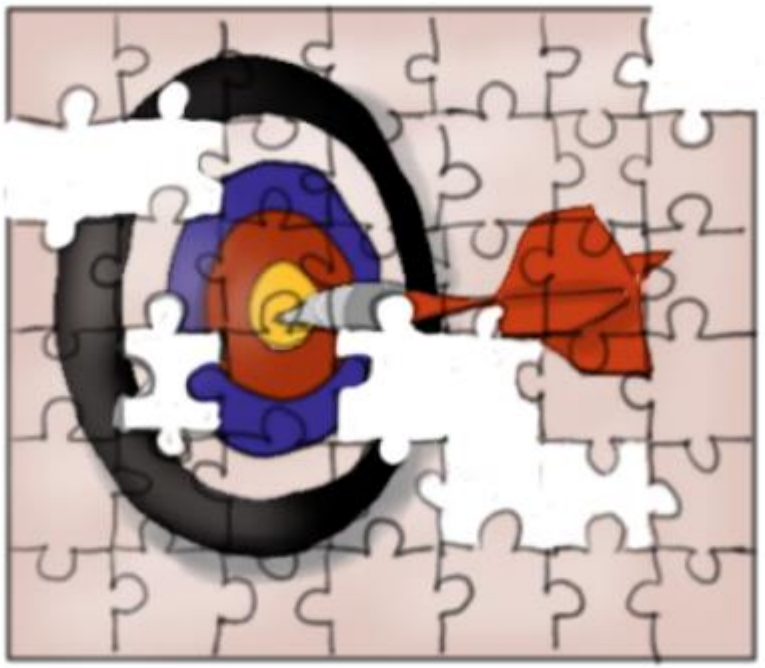
RE is mostly about identifying the purpose

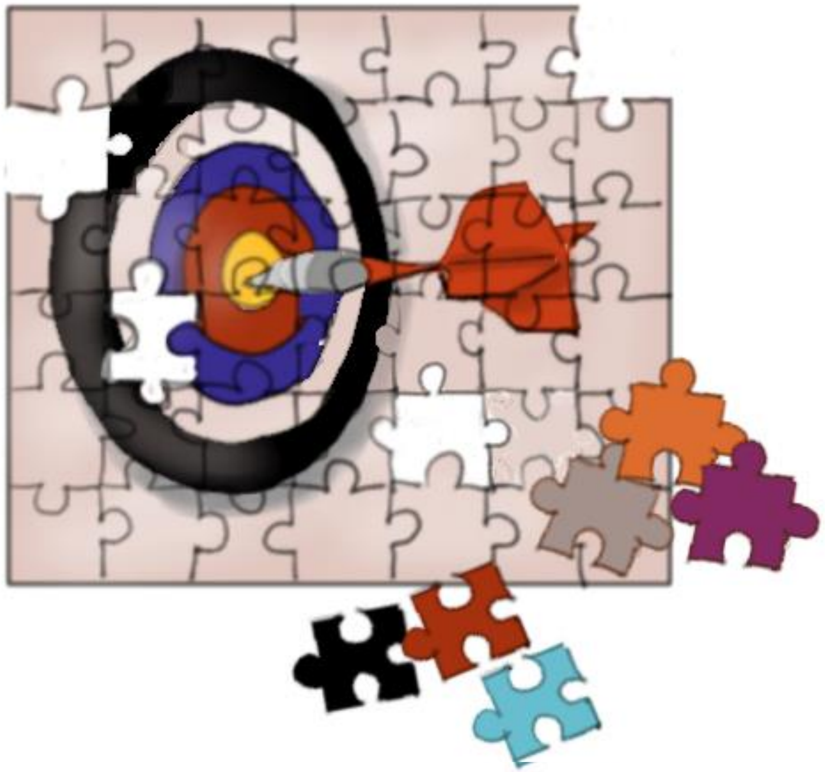
Identifying Purpose = Defining Requirements

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





Difficult to identify all requirements, incomplete requirements collected and S/W missing functionality

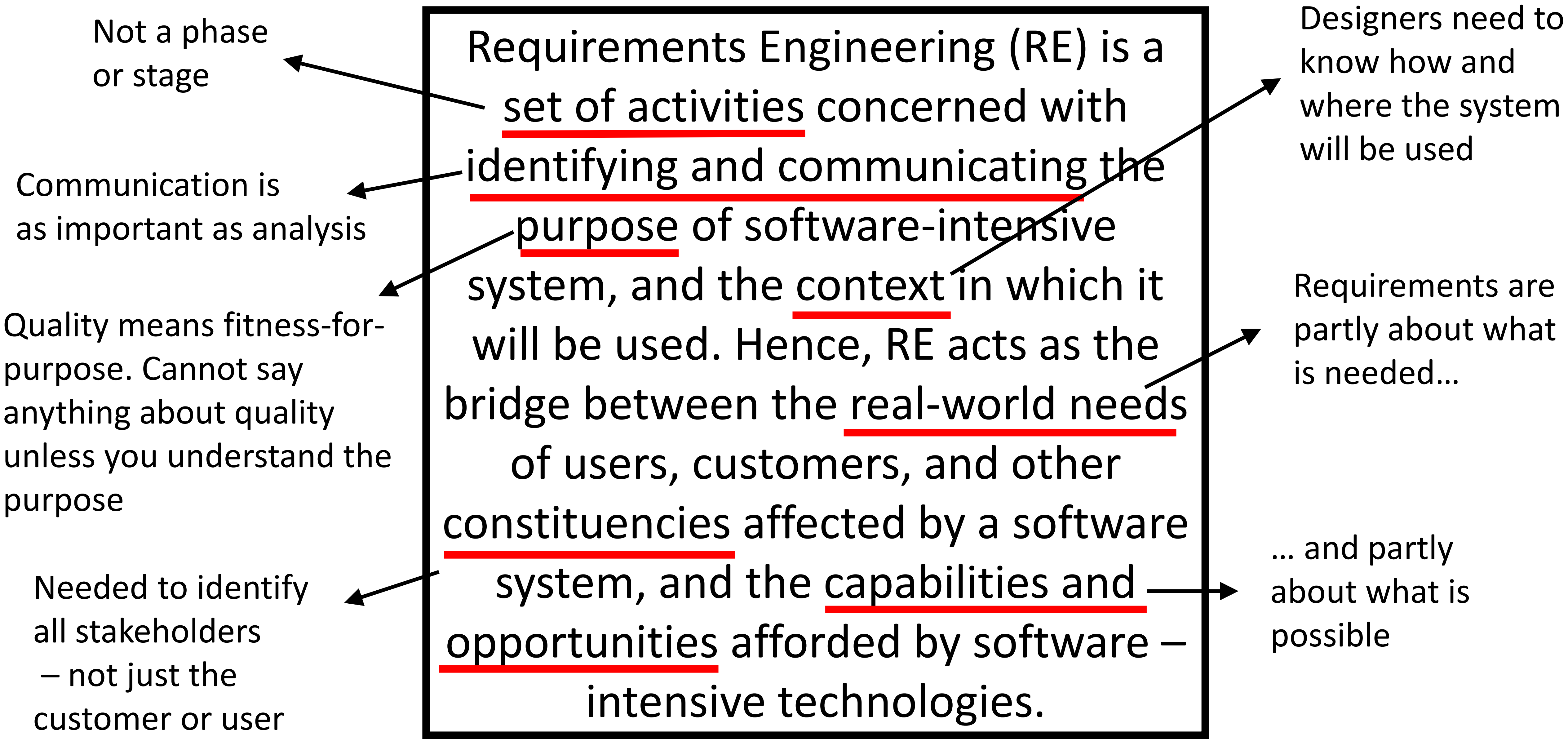
- Relevance of requirements;
- Irrelevant conflicting requirements collected for sake of completeness
- Worse case: completeness issue not solved, irrelevant requirements with information harmful to system

Best Practice?

PROJECT
12
9
3
6
TIC
TAC
TIC
TAC
BLA
BLA
BLA
BLA
BLA

- Identify a whole bunch of most obvious requirements
- Stakeholders sign off on them
- Problem: RE document long, unstructured, lot of information
- Not ideal, a rigorous and effective RE process

Definition of Requirements Engineering



Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of 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.

Not a phase or stage

Communication is as important as analysis

Quality means fitness-for-purpose. Cannot say anything about quality unless you understand the purpose

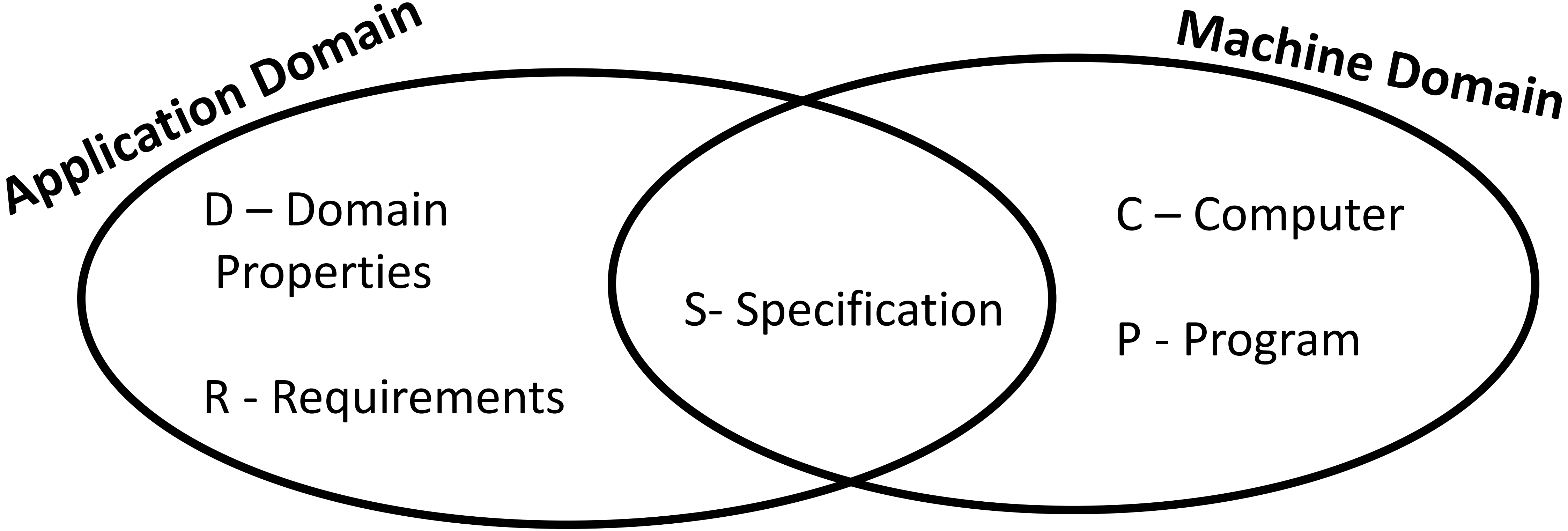
Needed to identify all stakeholders – not just the customer or user

Designers need to know how and where the system will be used

Requirements are partly about what is needed…

… and partly about what is possible

What are Requirements?



- Machine Domain - hardware/OS/libraries
- Application Domain - world in which software will operate
- Events in real world that machine can detect – buttons pushed
- Actions in real world that machine can cause- image appearing on screen

Functional and Non-functional Requirements

Functional

Non-Functional

Non-Functional requirements: refer to a system's non-functional properties such as security, accuracy, performance, cost, usability, adaptability, interoperability, reusability and so on.

User and System Requirements

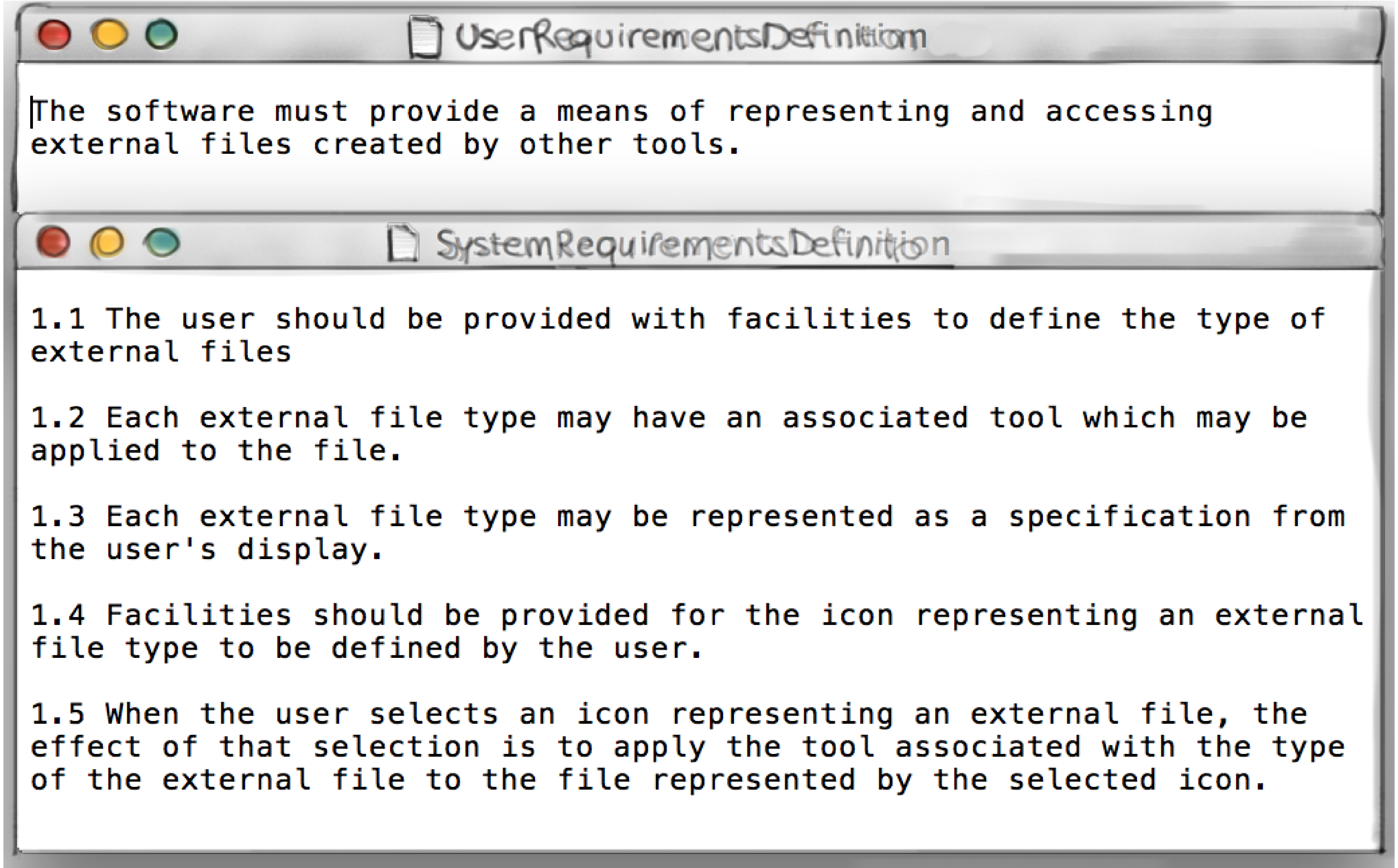
User Requirements

- 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 and System Requirements



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

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.

Where do Requirements come from?

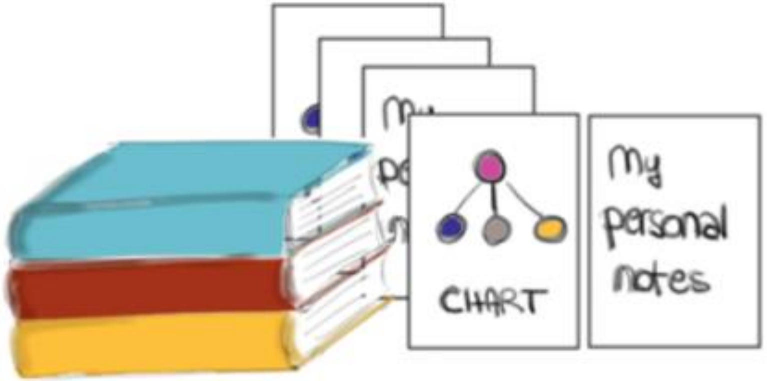
Stakeholders

...

Application Domain

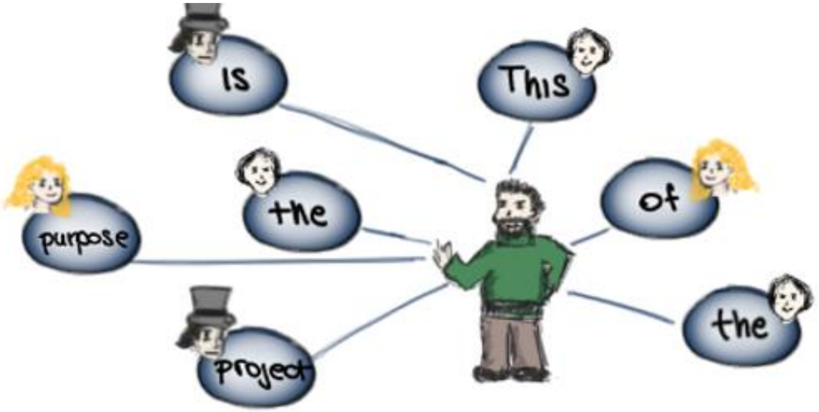
...

Documentation

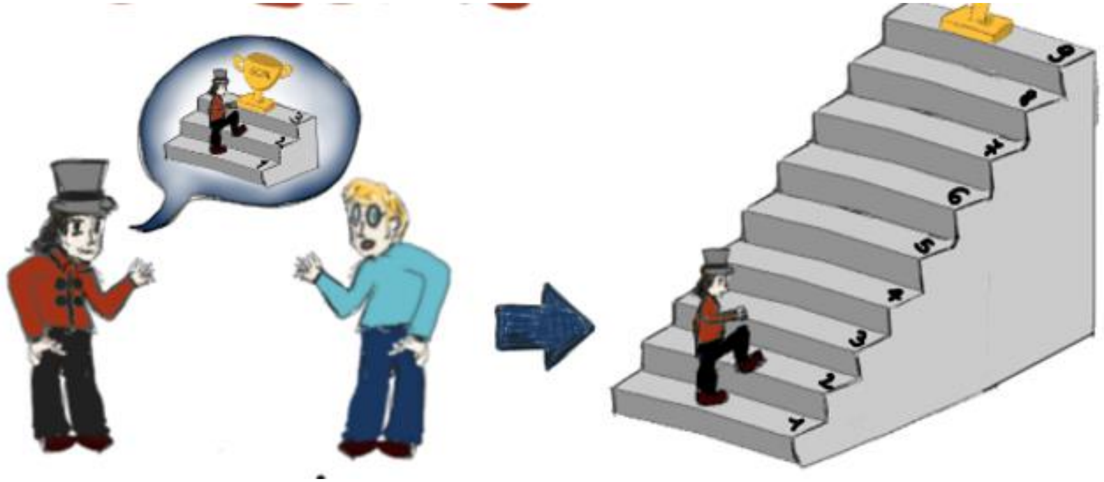


...

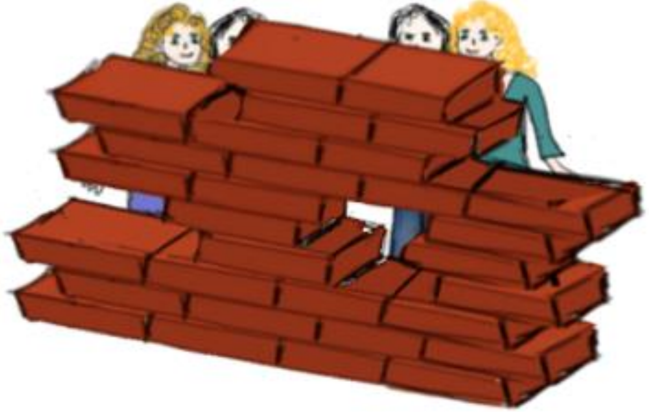
Elicitation Problems



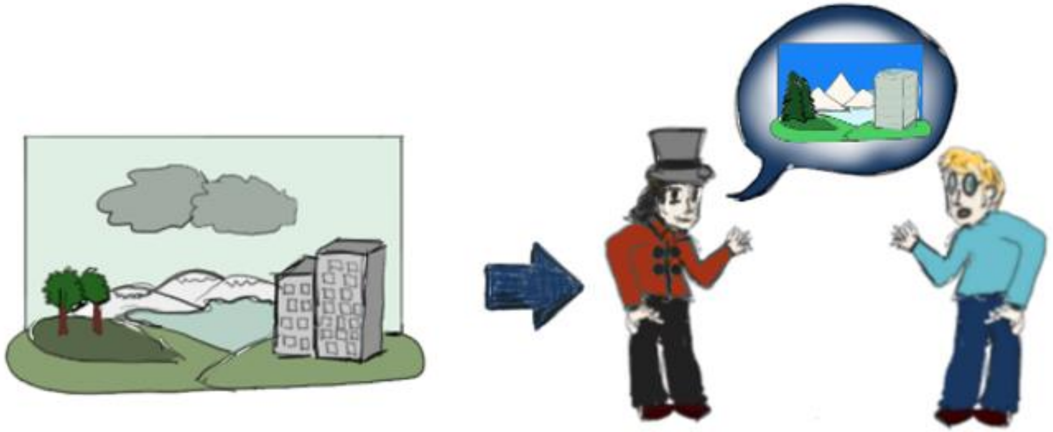
Thin spread of knowledge



Knowledge is tacit



Limited Observability



Bias

Traditional Techniques

Background Reading

Hard Data and Samples

Interviews

Surveys

Meetings

Other Techniques

Collaborative Techniques

brainstorming

Social Approaches

Ethnographic techniques

Cognitive techniques

Problem solving methods

Modelling Requirements

Modelling Enterprises

- Goals and objectives
- Organizational structure
- Tasks and dependencies
- Agents, roles, intentionality

Organization Modelling

Soft system modelling

Goal modelling

KAOS, CREWS

Modelling Information and Behavior

- Information Structure
- Behavioral View

– scenarios and use cases; state machine models, sequence diagrams, information flow

- Time/Sequencing requirements

Information Modelling (E-R , Class Diagram)

Structure Analysis (Structural Analysis & Design Technology)

Object Oriented Analysis (UML)

Formal methods (Alloy, PetriNet)

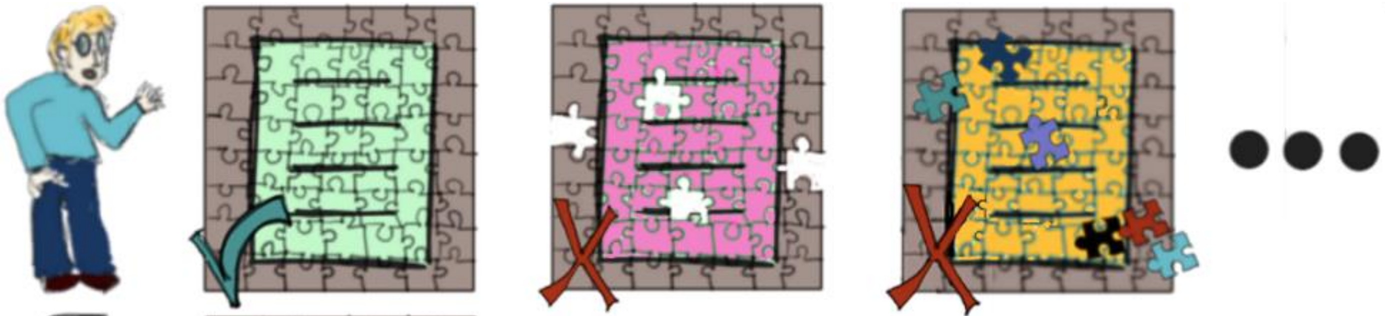
Modelling System Qualities (NFPs)

Quality Trade-offs (Win-win, NFR, AHP)

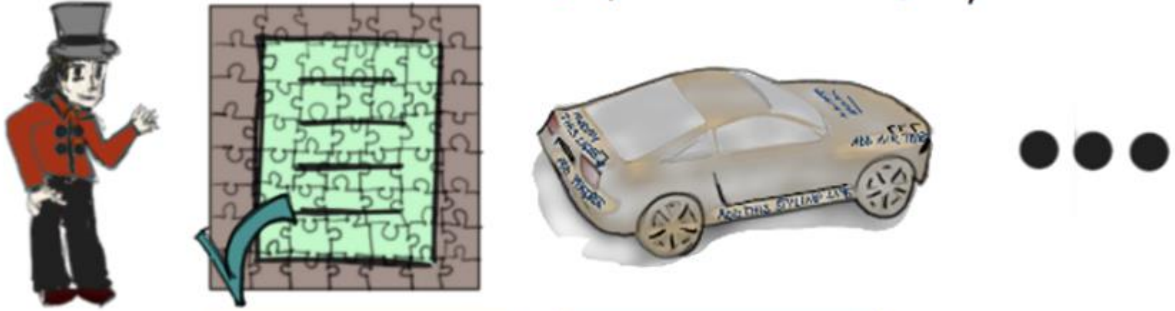
Specific NFPs (Timed PetriNet, Task Models)

Analyzing Requirements

Verification

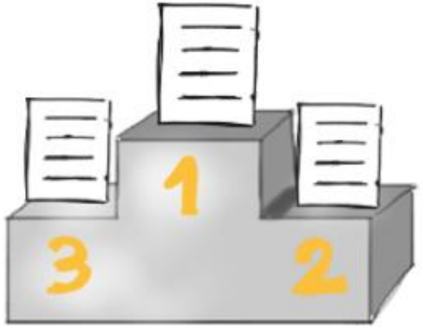


Validation



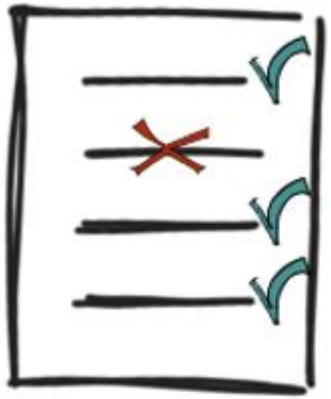
Risk Analysis

Requirements Prioritization



Limited Resources

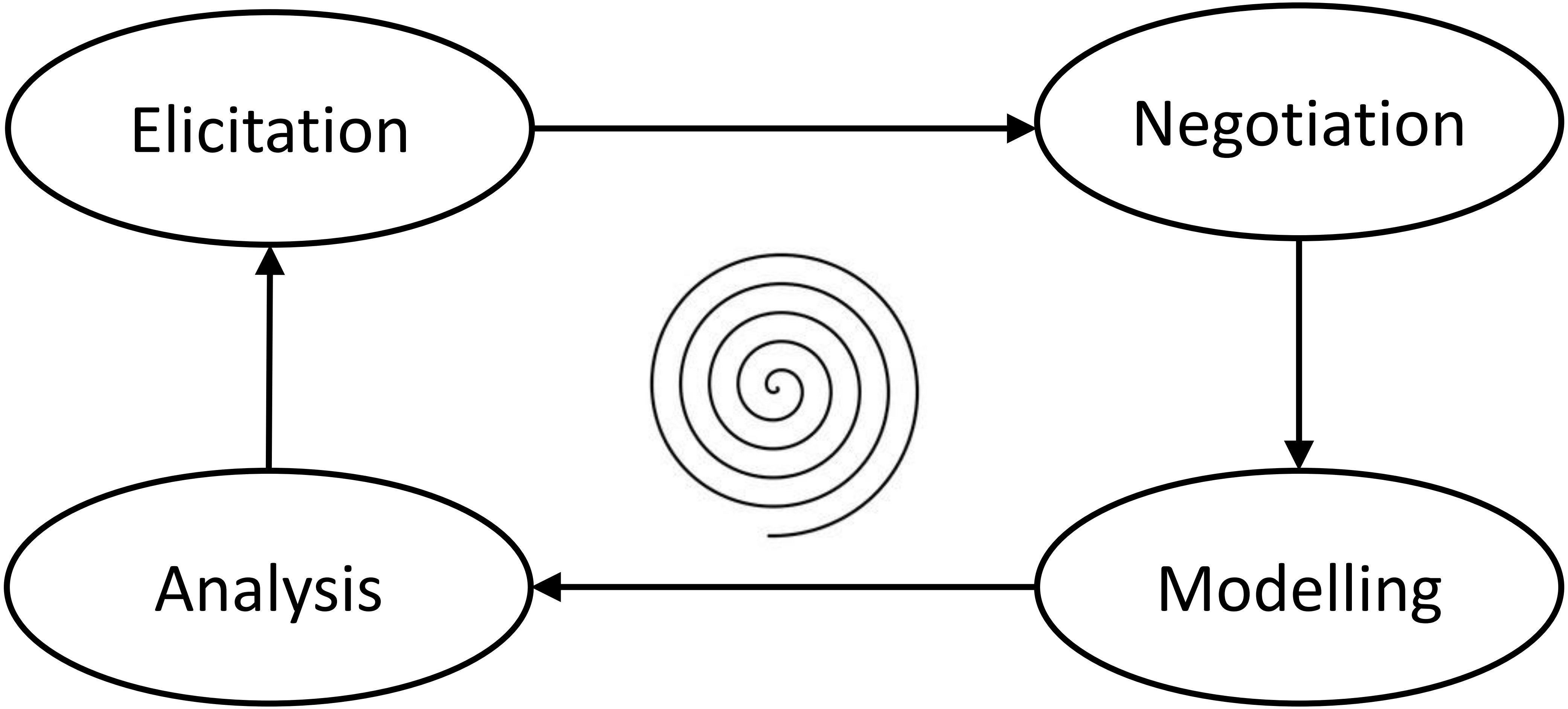
=> Inability to satisfy all the requirements



⇒Need to prioritize them

- Mandatory
- Nice to have
- Superfluous

Requirements Engineering Process



Elicitation

Negotiation

Modelling

Analysis