Announcements

- Quiz 2 will release tomorrow
  - Single attempt quiz with Honorlock; Duration of 10 minutes
  - Syllabus: Everything upto Requirements Engineering
- Project 1 Progress Report is Due tonight at 11:59 PM
  - 1 page document, 4 sections
- Requirements Assignment Out today for Project 1
- REST assignment was out this Tuesday- This is an individual assignment
  - Lecture video is on class website
- Extra Credit Opportunity in today’s class

CS3300 Introduction to Software Engineering

Lecture 06: Requirements Engineering

Dr. Nimisha Roy

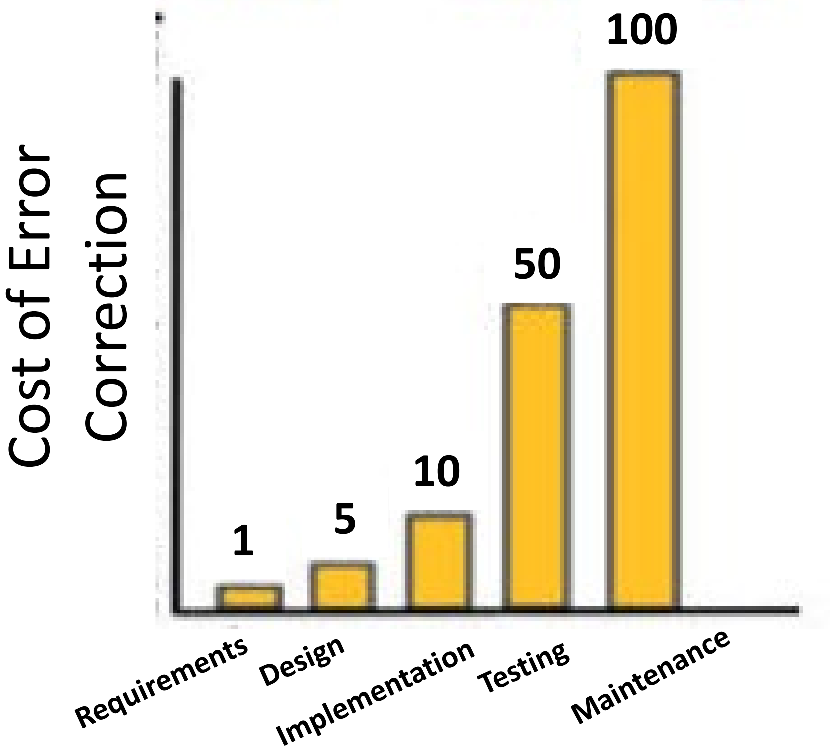
- nroy9@gatech.edu

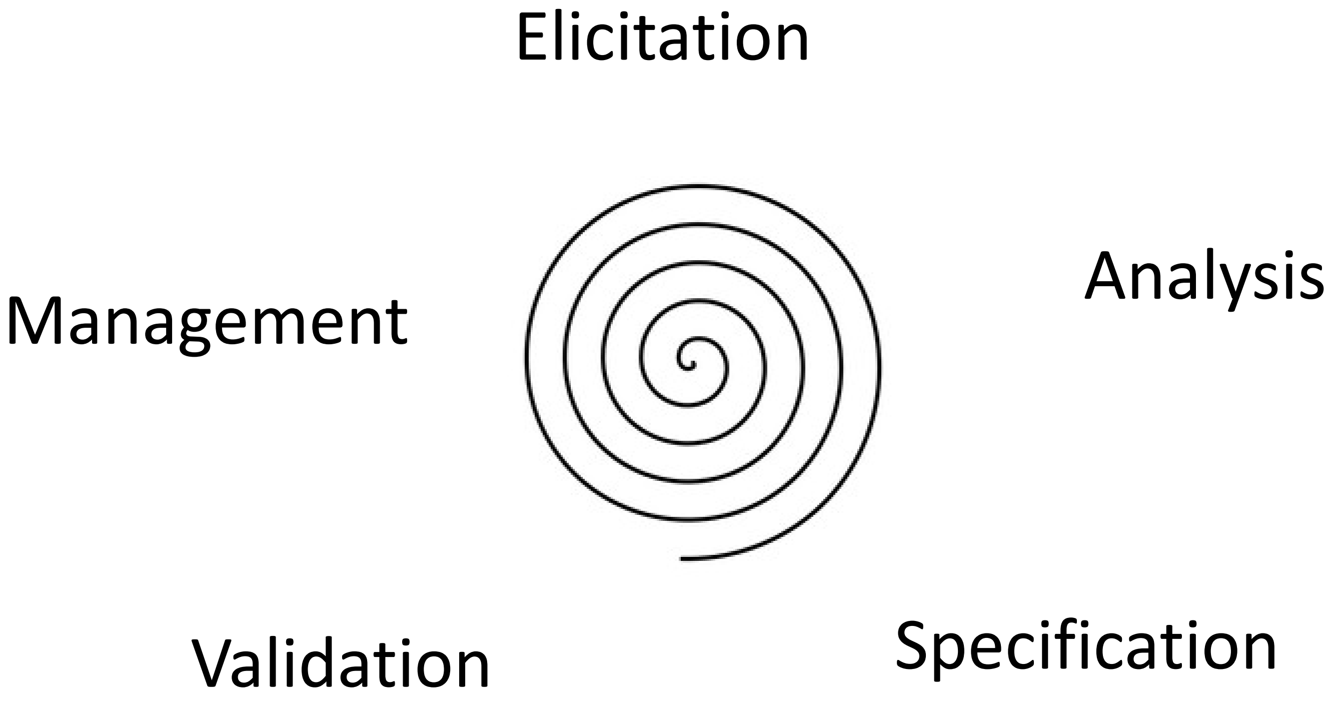
Slides adapted from Alessandro Orso

Requirements Engineering (RE)

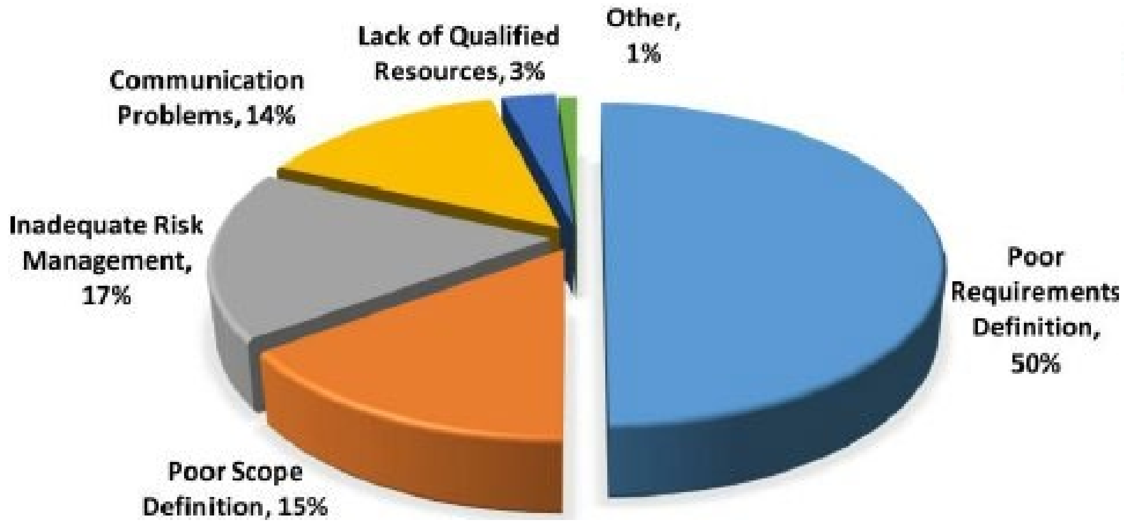
RE is the process of establishing the needs of stakeholders that are to be solved by software

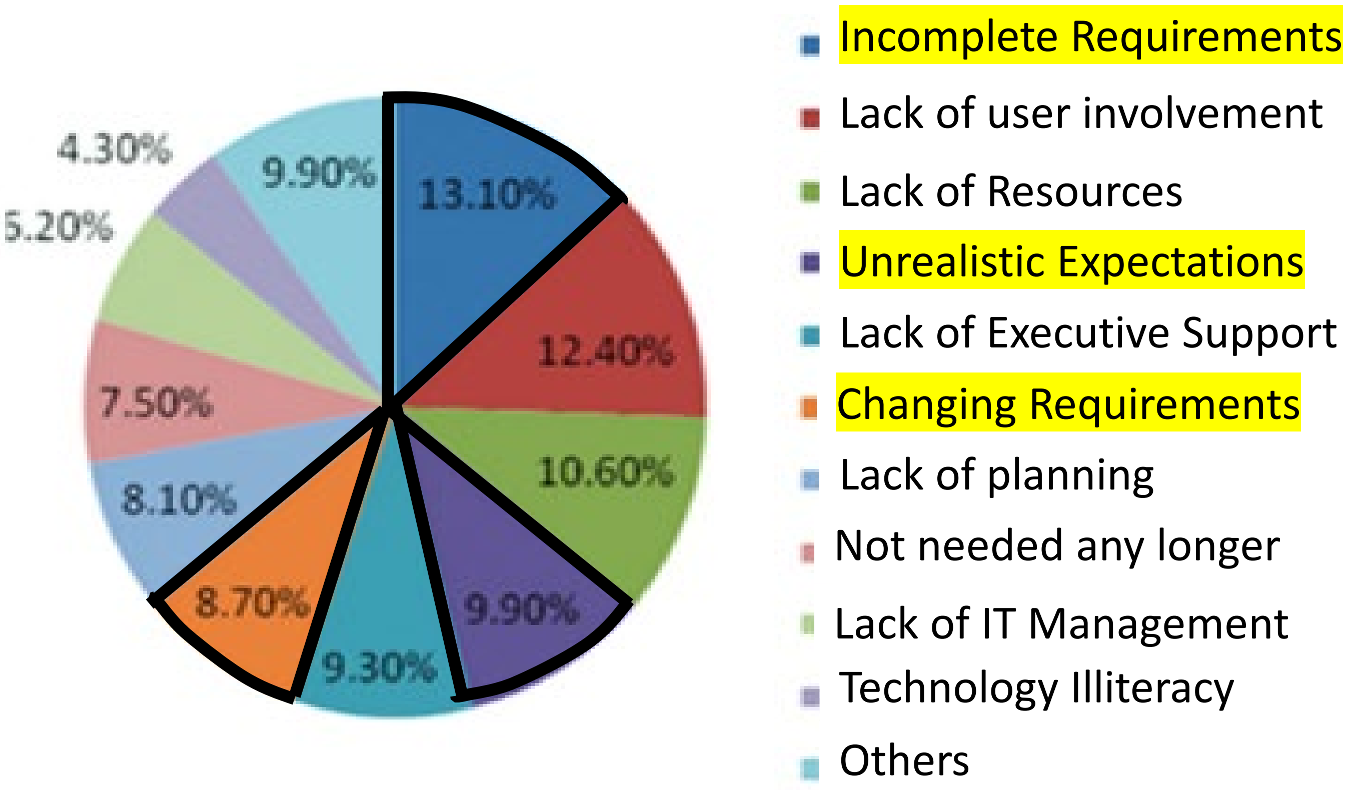
Cost of Late Correction





Software Failures due to RE





Role of Requirements in Software Project Failures (Agile)

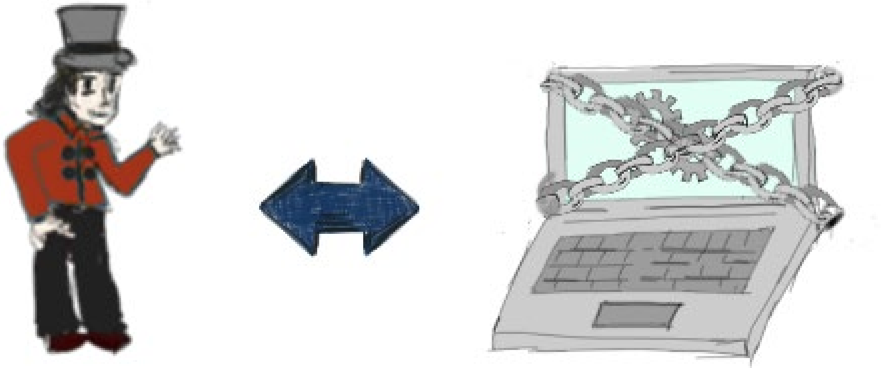
Source: Abdou, T., Kamthan, P., & Shahmir, N. (2014). User Stories for Agile Business: INVEST, Carefully!. Social Media and Publicity, 141.

Role of Requirements in Software Project Failures.

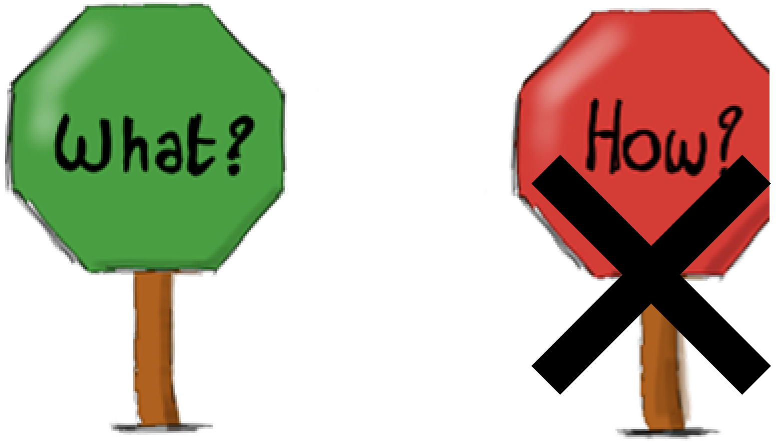
Source: Hussain, A., Mkpojiogu, E. O., & Kamal, F. M. (2016). The role of requirements in the success or failure of software projects. International Review of Management and Marketing, 6(7S), 306-311

Requirements Engineering (RE)

==

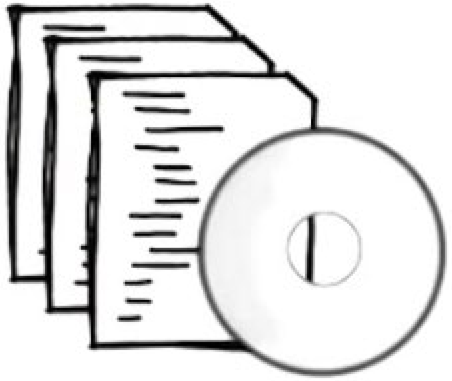


=> Software Requirements Specification (SRS)

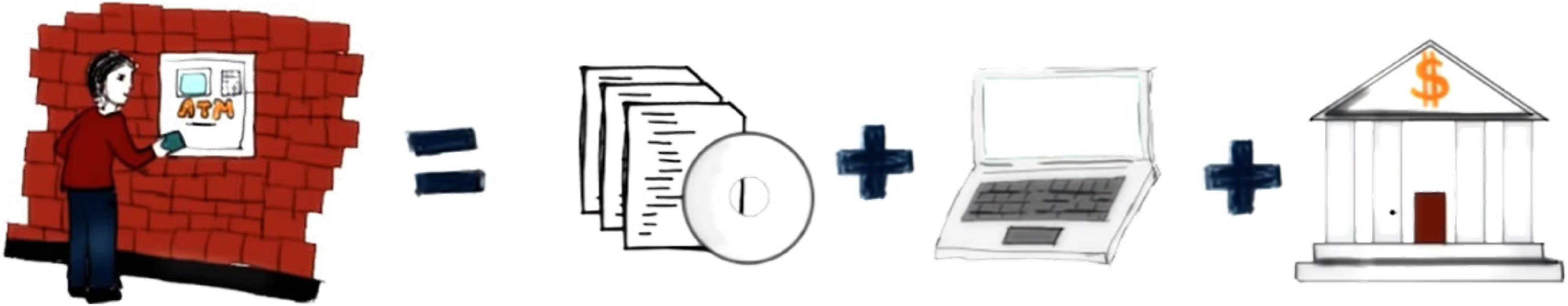


Software Intensive Systems

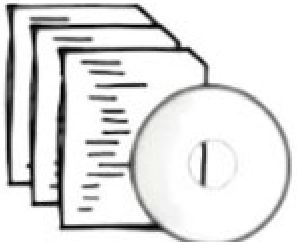
Software



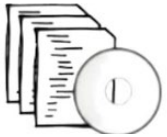
Software Intensive System = Software + Hardware + Context

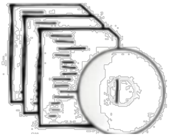
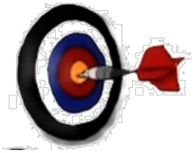


Software Quality



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” – Steve Jobs
- Changing requirements
- Multiple stakeholders with conflicting requirements

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.

Communication is as important as analysis

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

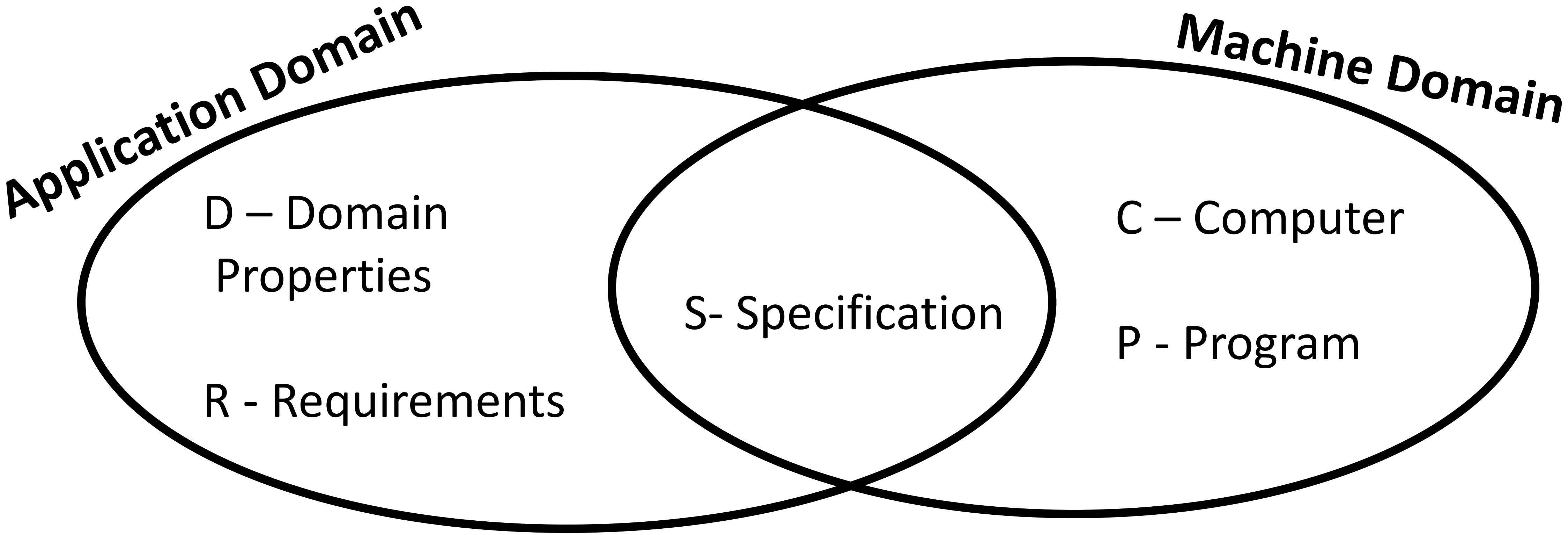
Needed to identify all stakeholders – not just the customer or user

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

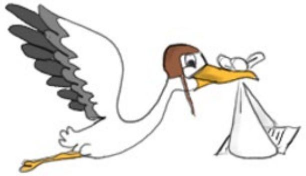
Requirements are partly about what is needed…

… 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



Requirements Elicitation: Where do Requirements come from?

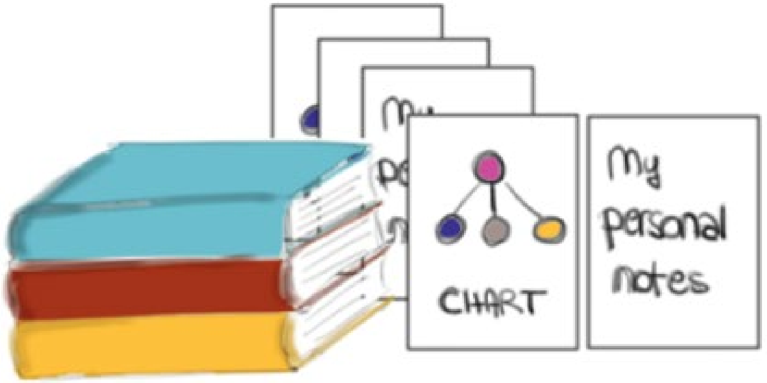
Stakeholders

...

Application Domain

...

Documentation



...

Requirements Elicitation: Other Techniques

Collaborative Techniques - Brainstorming

A software company is developing a new mobile game and invites both developers and gamers for a brainstorming session. During the session, a suggestion for a unique game mode emerges that becomes a central feature of the final product.

Social Approaches - Ethnographic techniques

Analysts spend a day in a busy restaurant to observe how servers use the point-of-sale system. They notice that servers often bypass a time-consuming screen, jotting orders down on paper to enter later, indicating a need for a more efficient user interface.

Requirements Elicitation: Other Techniques

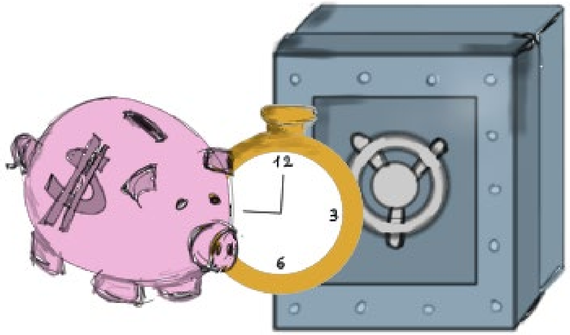
Cognitive techniques - Problem solving methods, Prototyping, Cognitive Task Analysis

In designing software for air traffic controllers, analysts study the controllers' processes for tracking multiple planes. They find that certain visual cues and alerts can reduce mental strain

Requirements Elicitation: Functional and Non-functional Requirements

Functional

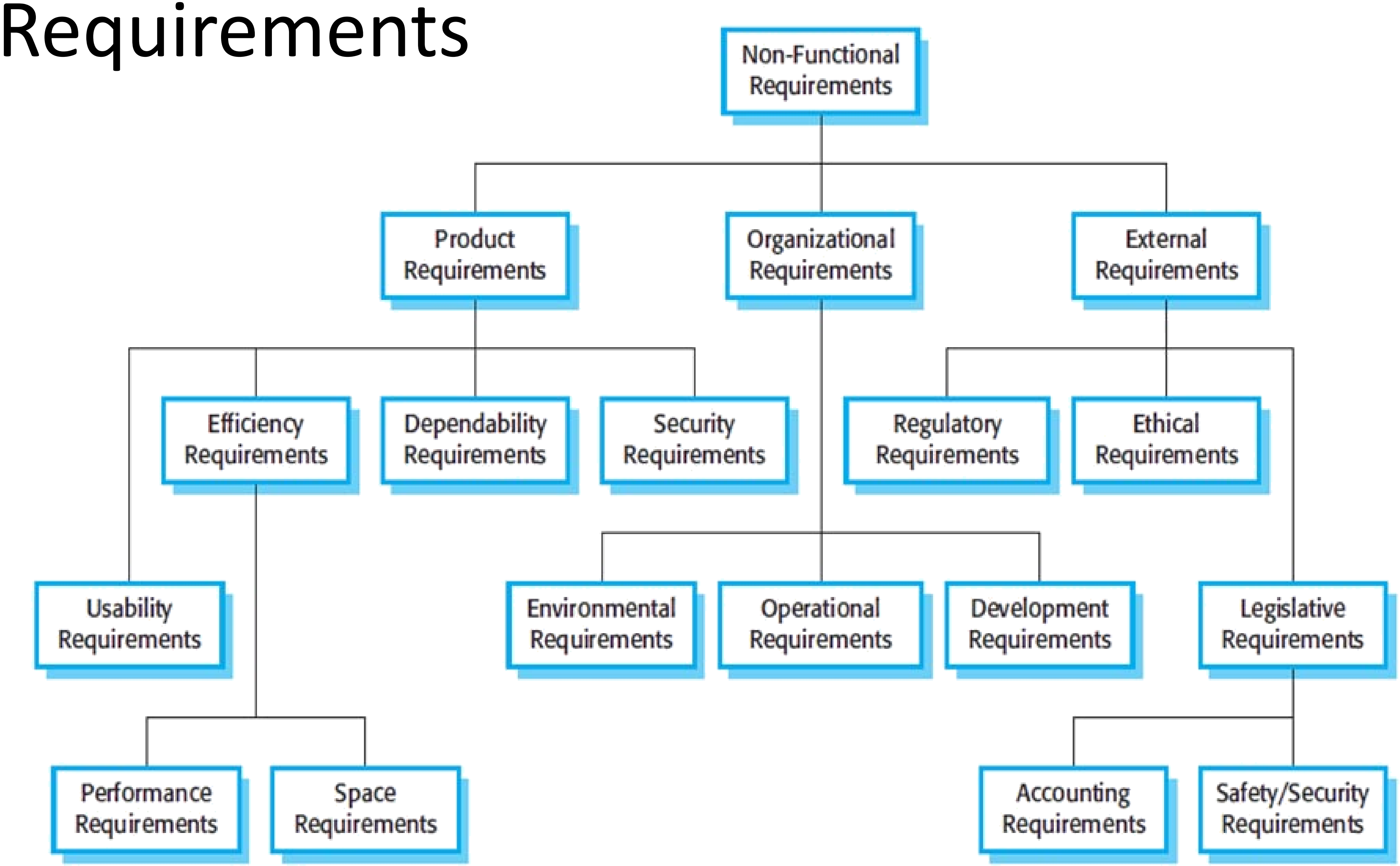
Non-Functional



Non-Functional requirements: criteria that can be used to judge the operation of a system, rather than specific behaviors. In essence, they describe how the system works, rather than what it does. E.g. security, accuracy, performance, cost, usability, adaptability, interoperability, reusability and so on.

Requirements Elicitation: Functional and Non-functional Requirements

Non-Functional Requirement may be more critical than functional requirement



An example: MHC-PMS: A patient management system for mental health care

- This case study describes a real system (although that is not its real name) which was used (and may still be used) in a number of UK hospitals, including hospitals in Scotland.
- The system is designed for use in clinics attended by patients suffering from mental health problems and records details of their consultations and conditions.
- It is separate from a more general patient records system as more detailed information has to be maintained and the system has to be set up to generate letters and reports of different types and to help ensure that the laws pertaining to mental health are maintained by staff treating patients.

MHC-PMS: Non functional Requirements

- ***Product requirement*** The MHC-PMS shall be available to all clinics during normal working hours (Mon–Fri, 0830–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.
- ***Organizational requirement*** Users of the MHC-PMS system shall authenticate themselves using their health authority identity card.
- ***External requirement:*** The system shall implement patient privacy provisions as set out in the country’s law.

Requirements Analysis: Traditional Techniques

Background Reading

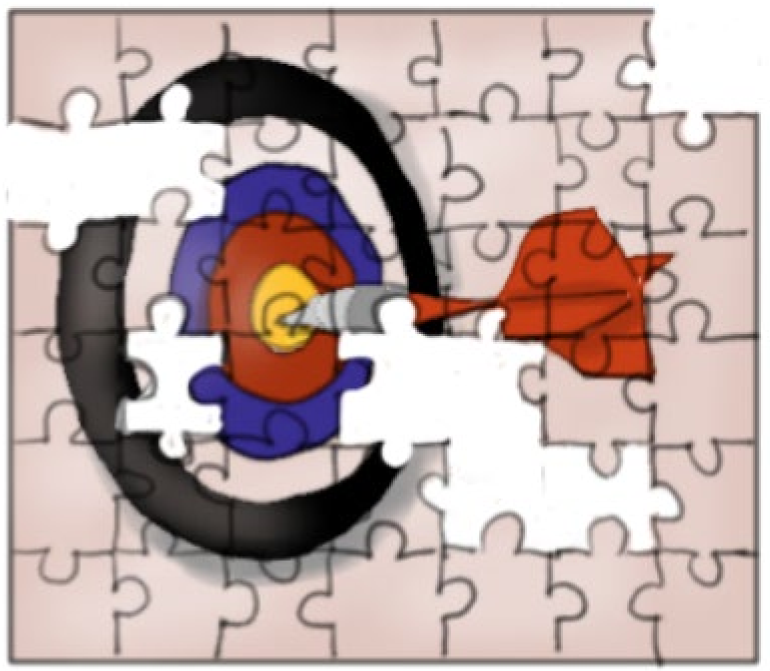
Hard Data and Samples

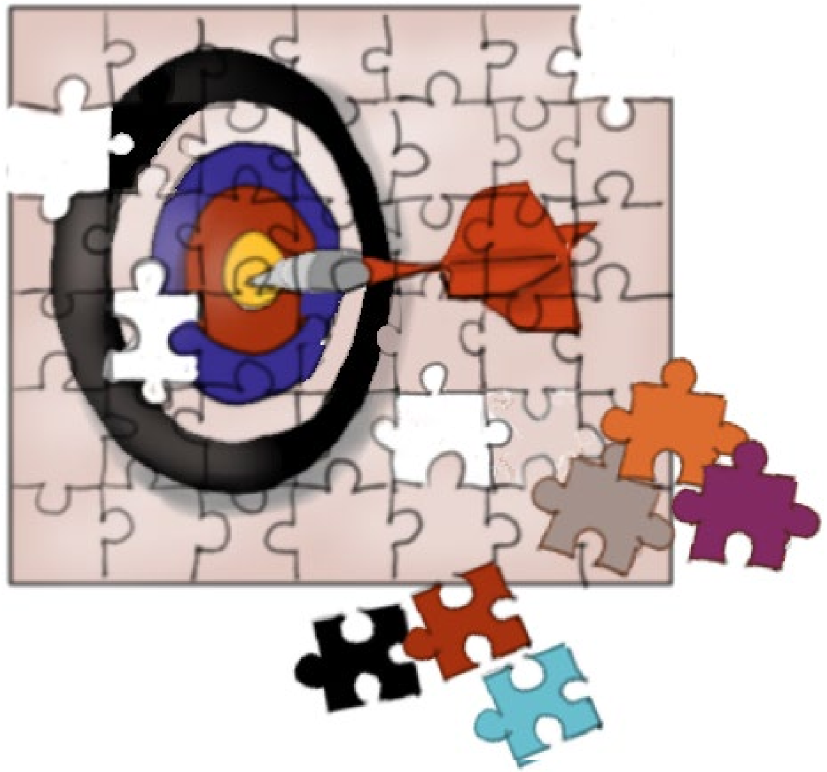
Interviews

Surveys

Meetings

Requirements Analysis: Completeness and Pertinence

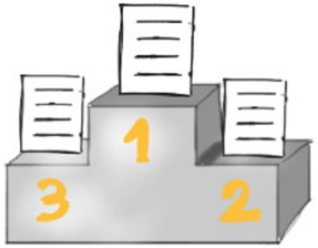




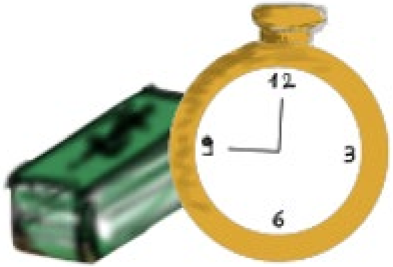
Difficult to identify all requirements, incomplete requirements collected and S/W missing functionality

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

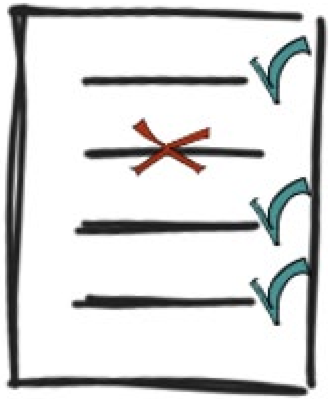
Requirements Analysis: Requirements Prioritization



Limited Resources



=> Inability to satisfy all the requirements



⇒Need to prioritize them

- Mandatory
- Nice to have
- Superfluous

Requirements Specification: User and System Requirements

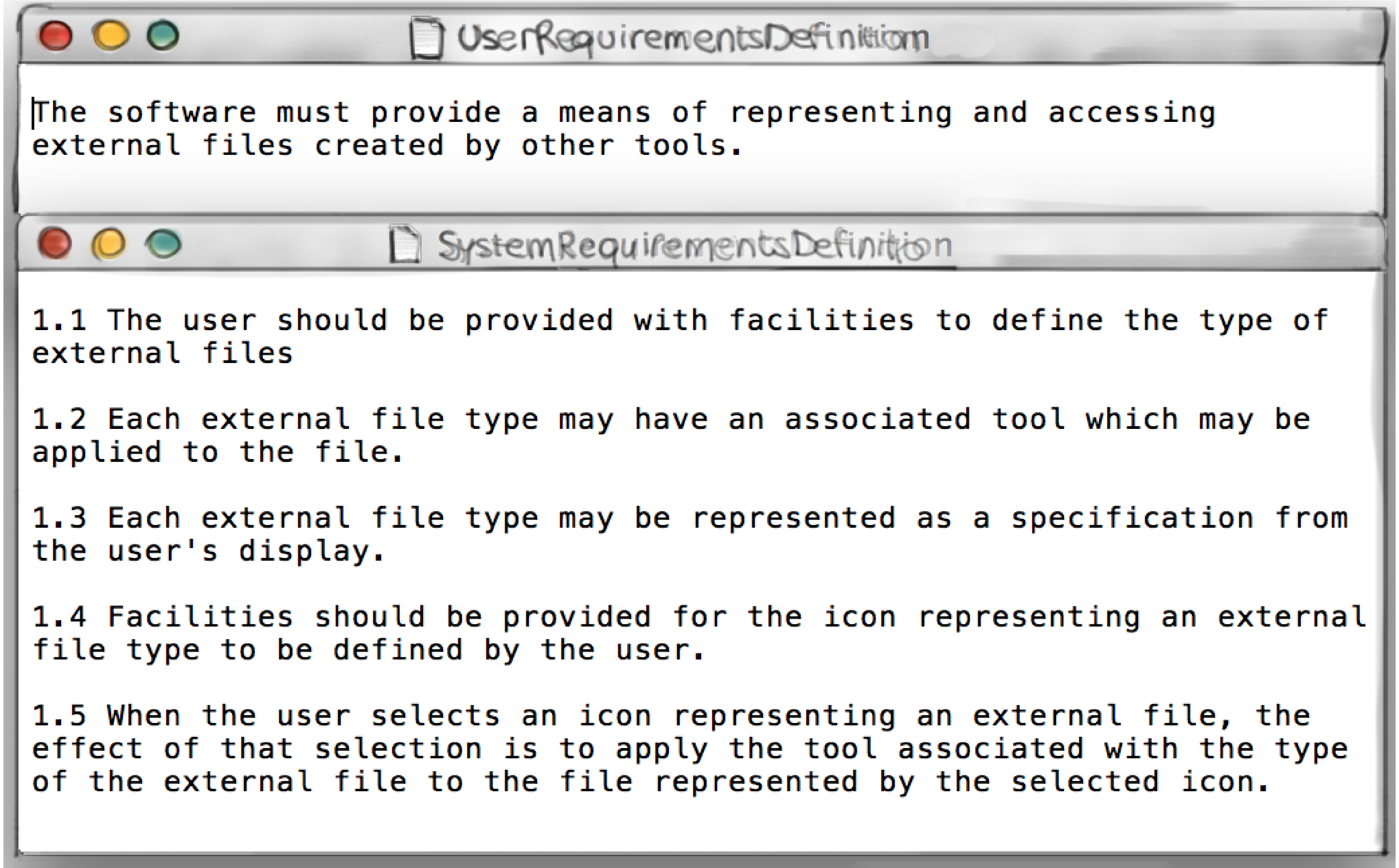
User Requirements

- 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.

Requirements Specification: Software specification

- The software requirements document is the official statement of what is required of the system developers.

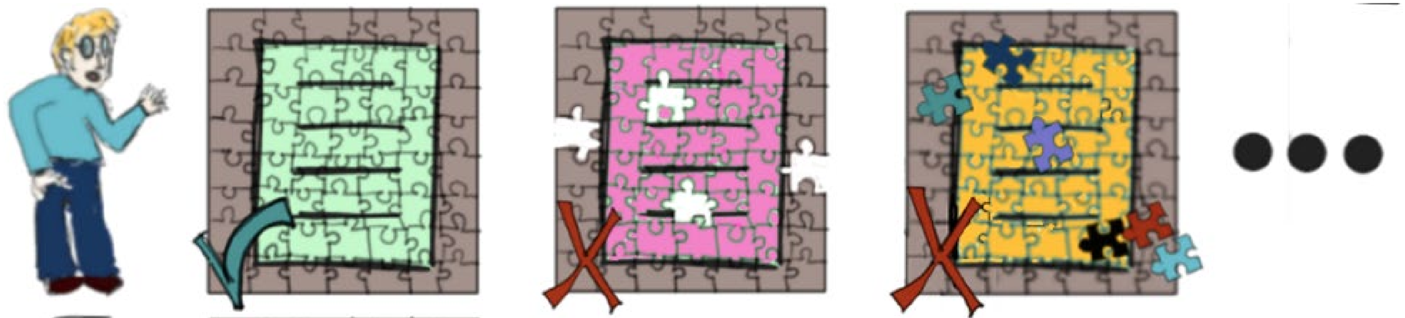
- Should include both a definition of user requirements and a specification of the system requirements.

- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it.

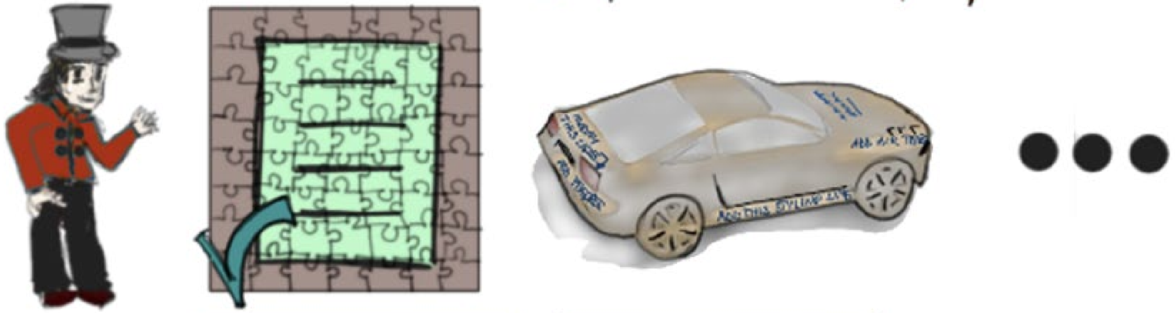
- Your project assignment uses the lightweight IEEE format of Software Requirements Specifications Document

Requirements Validation

Verification

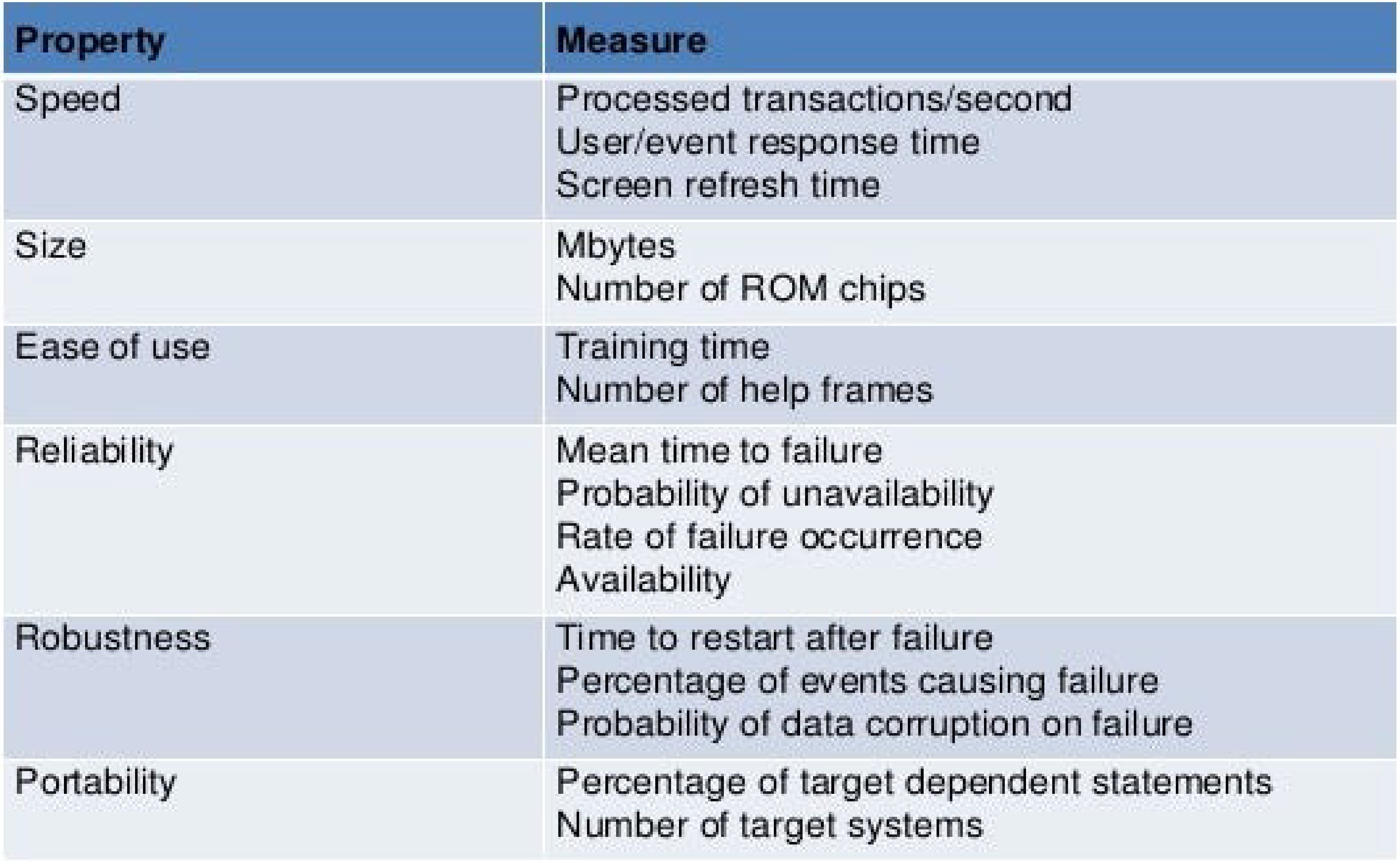


Validation



Risk Analysis

Requirements should be measurable

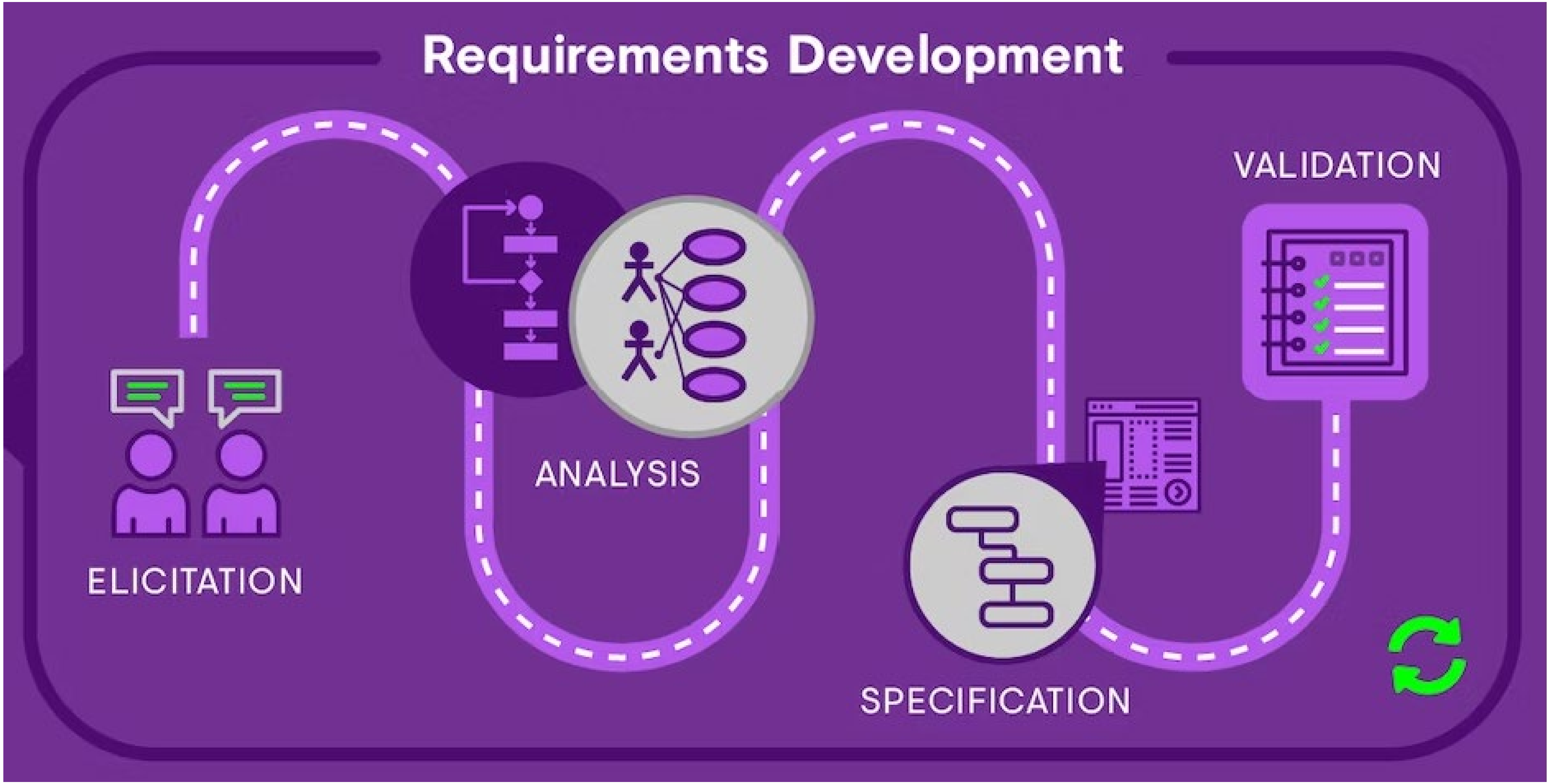


Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Requirements Engineering Process

1. Requirements Elicitation
    1. Consider functional and non functional requirements
2. Requirements Analysis
    1. User research
    2. Pertinence and Completeness
    3. Prioritize and Categorize
3. Requirements Specification
    1. User and system requirements
4. Requirements Validation
    1. Verification, Validation, Risk analysis
5. Management

Requirements Engineering Process



https://hackernoon.com/foo-xv1x3278

Quizizz