ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

# Software Testing and QA

Lecture 1-2

© Spring 2025 - **DR. Hesham Sakr**
Building, office No. 12
Hesham.sakr@sut.edu.eg

## Lectures:

|  |  |
|---|---|
| Wednesday : | 9:00AM – 10:40AM |
| Tuesday | 2:20PM- 4:00PM |

## COURSE OUTLINE ( What will you learn)

Introduction to Software Engineering

Software Quality - what is it, how is it measured, how is it achieved

Software process models

Requirement based quality assurance

Structured based quality assurance

Quality assurance documentation

Software Testing Systematic Testing - Black Box Testing - gray box testing

White Box Testing

Continuous Testing - defect testing

Automated testing –

Security and performance testing

## References

**Course notes:**

Based on selected chapters collected from text book.

**Textbook**:

Ian Sommerville, "Software Engineering", 9th edition

## Teaching and Learning Method

## The course will run as Project oriented course

1 Lectures

2 Project (running all the term)

3 Tutorials,.

## Assessment Method:

| | |
|---|---|
| 60% | Project, Assignments, Student presentation |
| 15% | Midterm Exam |
| 25% | Final Exam |

## Software products classification

### Generic products

Stand-alone systems that are marketed and sold to any customer who wishes to buy them (applications).

Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

### Customized products

Software that is commissioned by a specific customer to meet their own needs.

Examples – embedded control systems, air traffic control software, traffic monitoring systems.

## What are the attributes of good software?

**Good software should deliver the required functionality user.**

**Also the software should include another properties such as (performance, flexibility maintainability, reliability and security).**

## Essential attributes of good software

| Product characteristic | Description |
|---|---|
| **Flexibility, Maintainability** | Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment. |
| **Reliability and security** | Software dependability includes a range of characteristics including reliability, security and safety. Software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system. |
| **Efficiency** | Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc. |
| **Acceptability** | Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use. |

## Functional and nonfunctional properties of software

✧ Functional properties

- The attributes which are concerned with the functionality offered by the software such as ( searching, creating tables, updating files)

✧ Nonfunctional properties

● The attributes which are not concerned with the functionality Such as (efficiency- security- cost- reliability- maintainability- changeability).

## Software process activities

| | |
|---|---|
| Software Specification | Analyses, Requirements |
| Software Design | Development |
| Software implementation | |
| Software testing | Validation |
| Software Evolution | Upgrading changes |

## The software process

- A structured set of activities required to develop a software system.
- Many different software processes but all involve:
  - Specification: defining what the system should do;
  - Design: defining the organization of the system
  - Implementation: coding the system;
  - Validation: checking that it does what the customer wants;
  - Evolution: changing the system in response to changing customer needs.

## The software process models

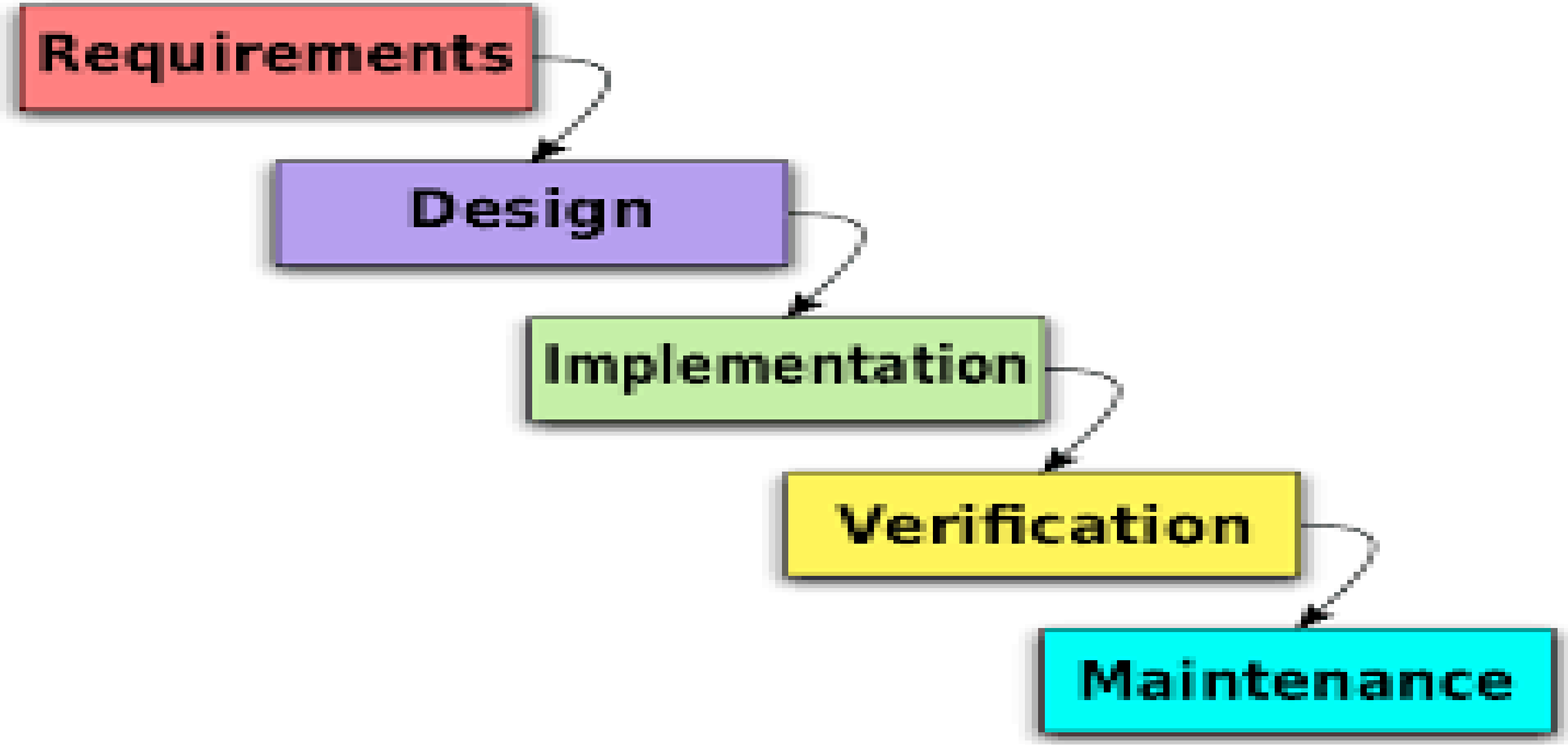
- A software process model represents the order in which the activities of software development will be undertaken.

- It describes the sequence in which the phases of the software lifecycle will be performed.

## The software process models

- Waterfall model
- Agile model
  - RAD (rapid application development)
  - Incremental model
- DevOps
  - a set of practices that combines the abilities of Software Development i.e Dev and IT Operations i.e Ops together, which results in delivering top- notch quality software fastly and more efficiently

## Waterfall model



## Waterfall model

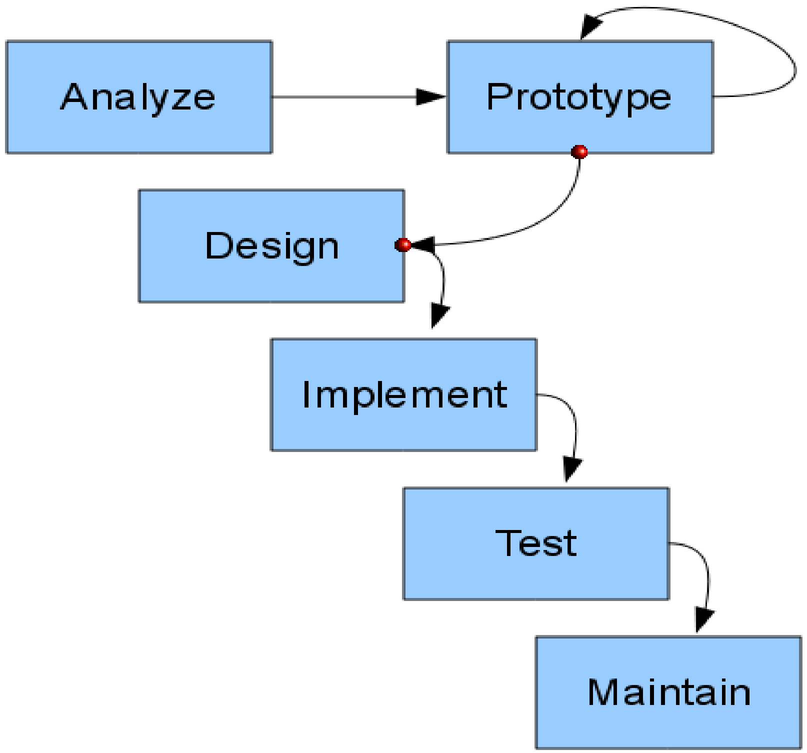
**Advantages:**

- Simple, easy to understand and follow. Highly structured.
- After specification is complete, low customer involvement required.

**Disadvantages:**

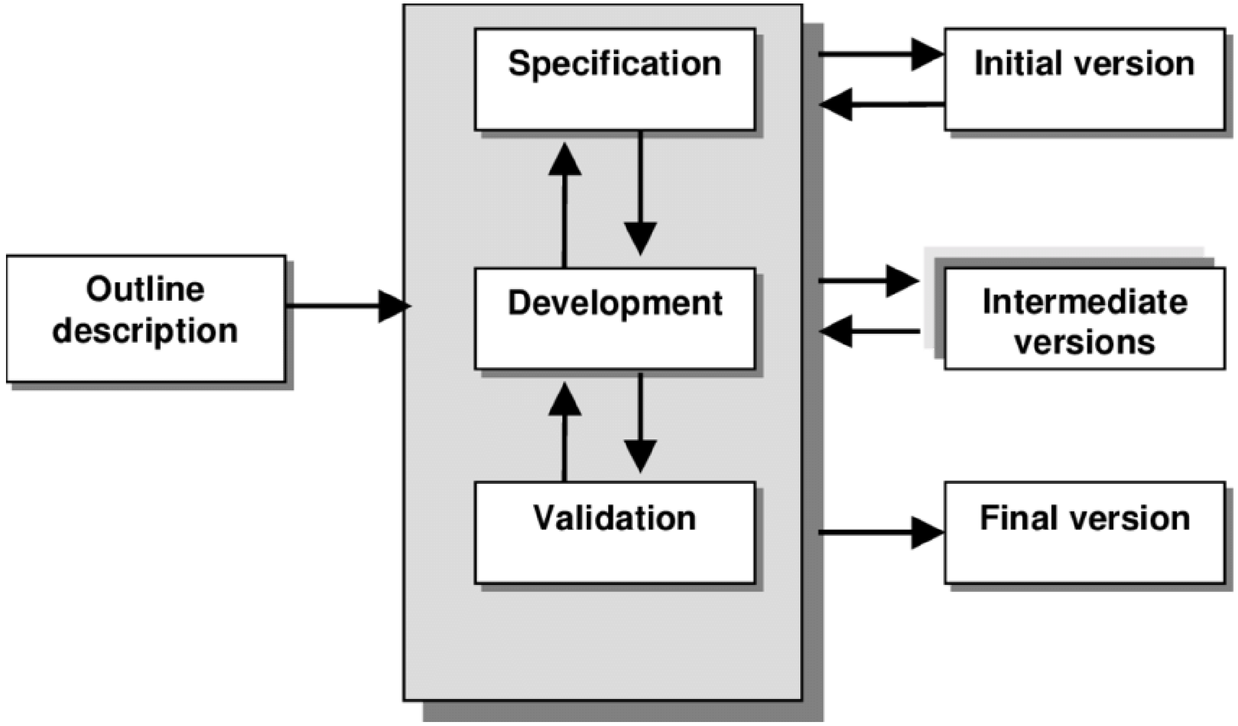
- Inflexible - can't adapt to changes in requirements.

## Waterfall with Prototyping Model



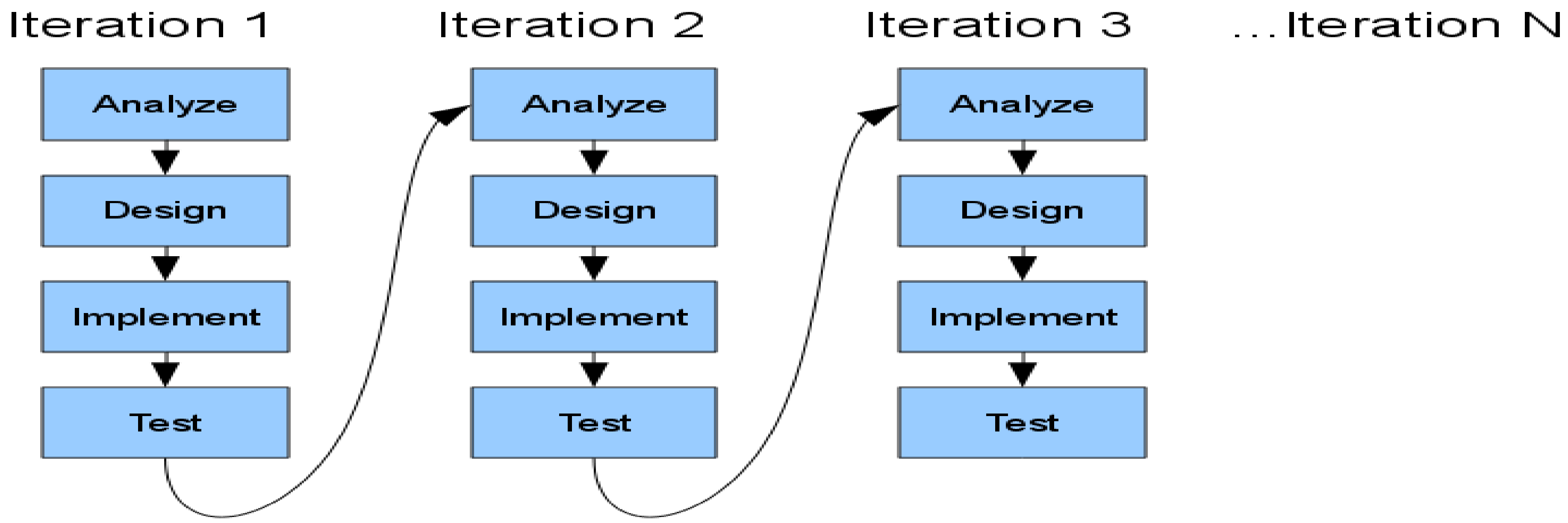
- A variation of the waterfall that adds a new phase, prototyping.
- It can also be used to better understand and elicit user requirements

## Incremental Model



## Agile Model

An iterative approach. During each iteration a single feature or small set of features are chosen and implemented completely



## Agile Model

Can adapt to changing requirements because you haven't committed to big design that encompasses everything.

Requires huge customer involvement.

Suitable for evolving several teams. Allow rapid development.

## Extreme Programming (XP)

- XP teams practice test-driven development technique (TDD)
- Writing an automated unit test before the code itself.
- Pair Programming
- Two programmers work jointly on the same code.
- The first developer focuses on writing, the other one reviews code, suggests improvements, and fixes mistakes along the way.

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

## Requirement engineering

## Requirements and design

Requirements should state what the system should do.

System design should describe how it does this.

## Requirements engineering process

- **Feasibility study**

Is it technically and financially feasible to build the system?

- **Requirements elicitation and analysis**

What do the system stakeholders require or expect from the system?

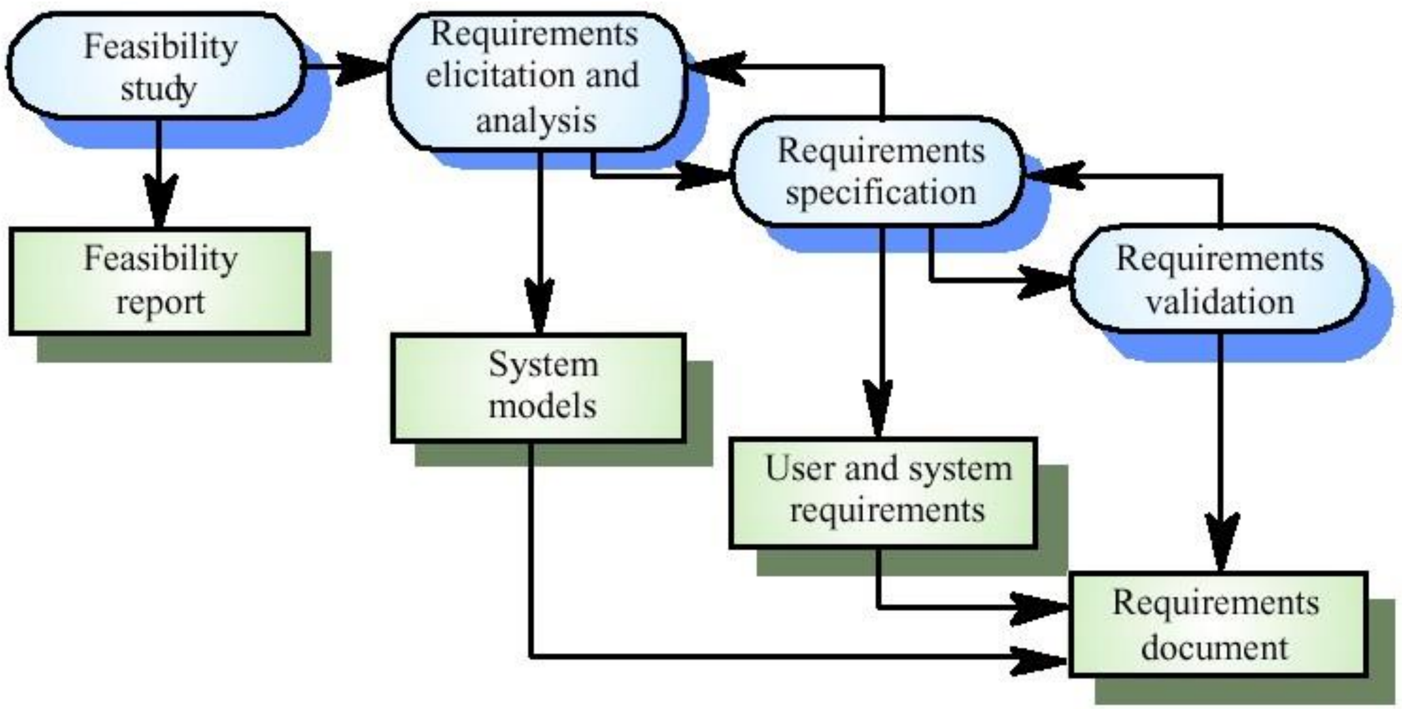
- **Requirements specification**

Defining the requirements in detail

- **Requirements validation**

Checking the validity of the requirements

## The requirements engineering process



## Requirements and analyses

- Stockholders Requirements: restrict access of the system to definite users.
- Formal Requirement: Users Authentication
- Analyses : who is the user

What does the user know

## Requirements and analyses

Authentication can be done by

Entering a password,

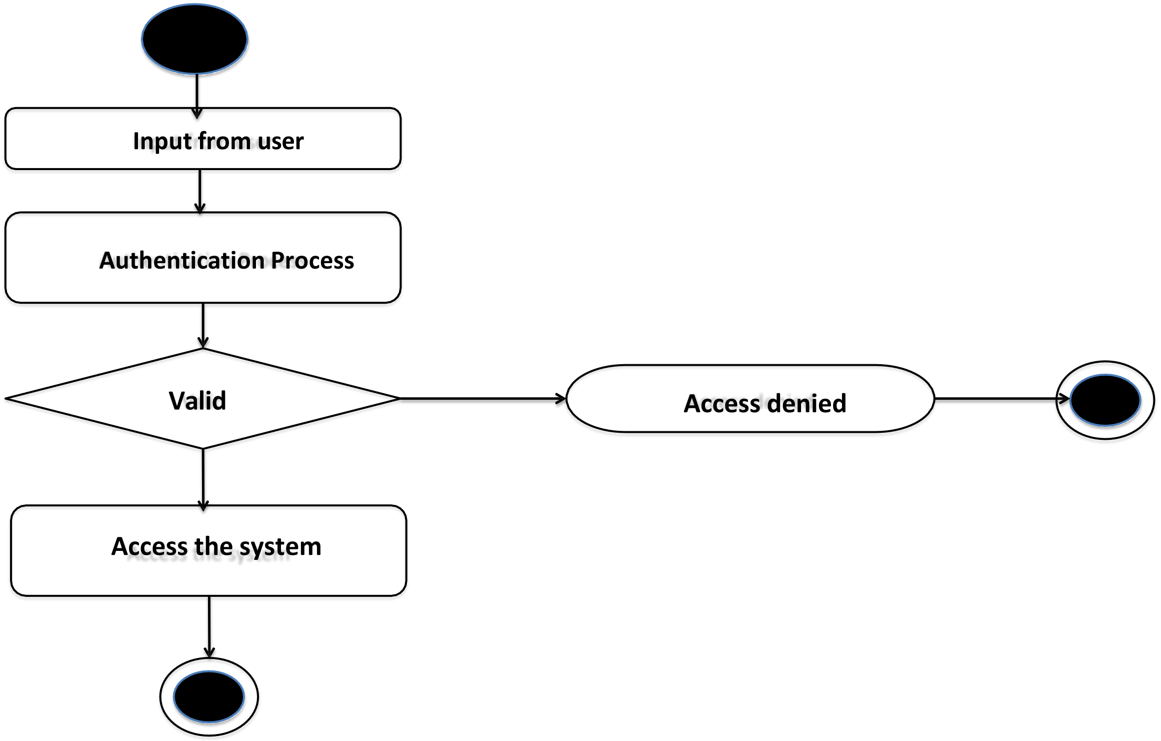
Inserting a smart card and entering the associated PIN,

Providing a fingerprint,

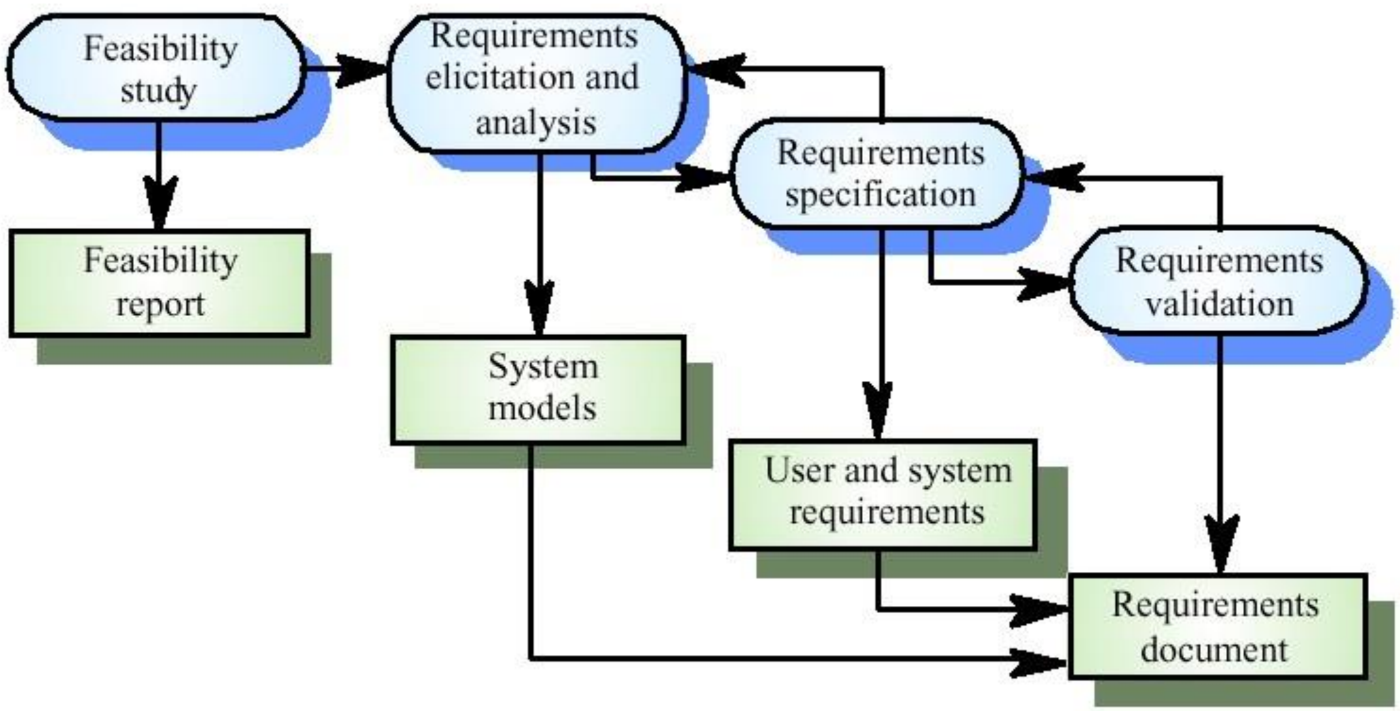
Voice pattern sample,

Retinal scan

## Authentication Process



## The requirements engineering process



## Feasibility studies

- A feasibility study decides whether or not the proposed system is worthwhile

- A short focused study that checks:
  - If the system contributes to organisational objectives
  - If the system can be engineered using current technology and within budget
  - If the system can be integrated with other systems that are used

## Feasibility study implementation

- Based on information assessment (what is required), information collection and report writing
- Questions for people in the organisation
  - What if the system wasn’t implemented?
  - What are current process problems?
  - How will the proposed system help?
  - What will be the integration problems?
  - Is new technology needed? What skills?
  - What facilities must be supported by the proposed system?

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

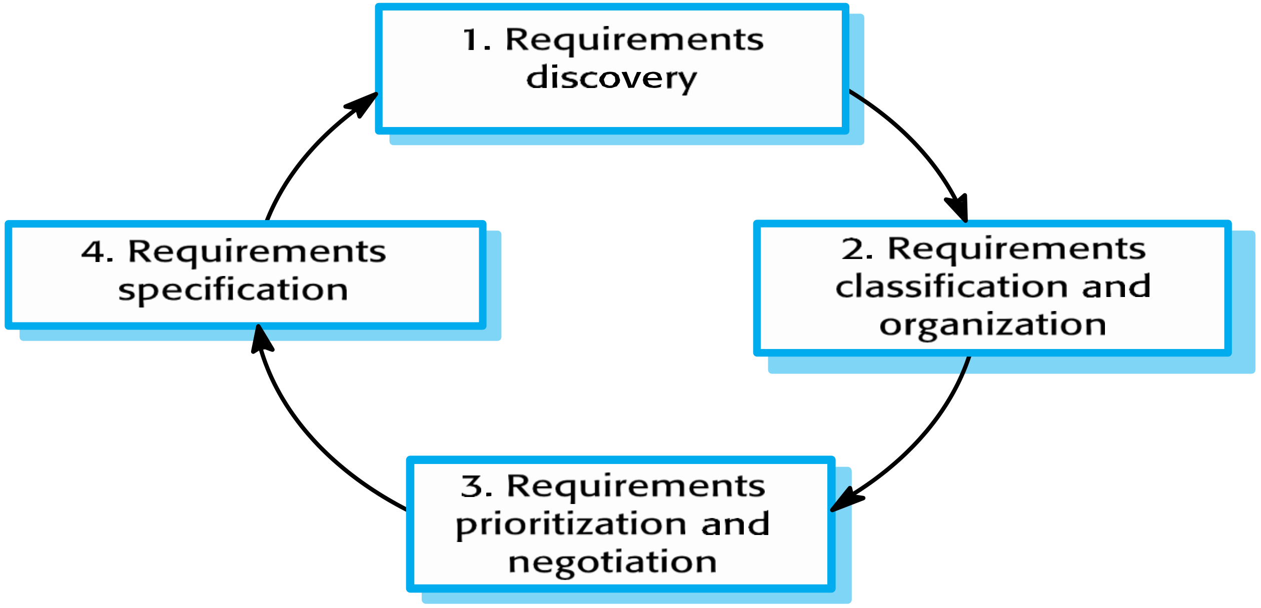
## Requirements elicitation (specification)

## Requirements elicitation and analysis

- Sometimes called requirements elicitation or requirements discovery.
- Involves technical staff working with customers to find out the services that the system should provide and the system’s operational constraints.

- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc.

## The requirements elicitation and analysis process


1. Requirements discovery
2. Requirements classification and organization
3. Requirements prioritization and negotiation
4. Requirements specification

## Requirements discovery

The process of gathering information about the required and existing systems and distilling the user and system requirements from this information.

Interview, questionnaires, ethnography.

## Mechanism

- A social scientist spends a considerable time observing and analysing how people actually work.

- People do not have to explain or articulate their work.

- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Thank you