Introduction to Information Systems Analysis and Design (ISAD)

Learning Objectives

- Explain the role of ISAD in developing effective information systems.
- Compare traditional SDLC vs. modern Agile approaches.
- Recognize project management principles in IS projects.

Content

- Concept of Information Systems
- Overview ISAD
- Systems Development Life Cycle (SDLC)
- Software Development Life Cycle
- Function-Oriented and Object-Oriented Approaches

Concept of Information Systems

- **Information**
  - is a type of **organizational resource** that must be **carefully managed**, just like any other resource.
  - Processing information requires **costs in terms of time, money, and human effort**.
  - Therefore, information processing should aim to **maximize the potential value** of this resource.

Concept of Information Systems

- **Information System (IS)**
  - An **information system** is a system that consists of **people, data, processes, and information technology** that **interact with one another** to **collect, process, store, and provide** necessary information outputs in order to **support a particular system or organization**.
  - Information systems exist in **various forms and scales** across all types of organizations.

Classification of Information Systems

Information systems can be classified according to the functions they serve:

- **Transaction Processing System (TPS):** An information system that collects and processes data about business transactions.
- **Management Information System (MIS):** An information system that provides management-oriented reports based on the processing of transaction data and organizational operations.
- **Decision Support System (DSS):** An information system that assists in identifying decision opportunities and provides information to support the decision-making process.
- **Executive Information System (EIS):** An information system that supports the planning and evaluation needs of senior executives.

Classification of Information Systems

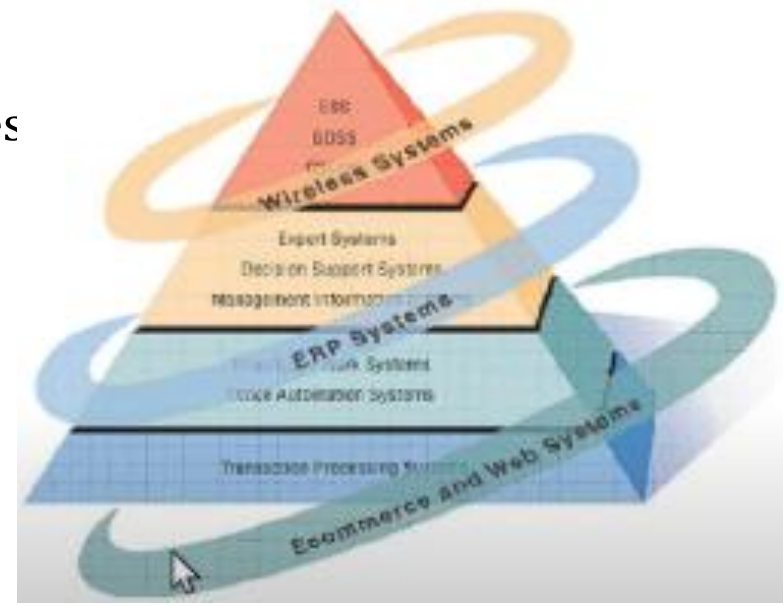
Information systems can also be classified according to the functions they serve:

- **Expert System (ES):** An information system that captures the specialized knowledge of experts and simulates that knowledge to provide benefits to ordinary users.
- **Communication and Collaboration System:** An information system that enhances communication efficiency among employees, partners, customers, and suppliers, thereby strengthening collaboration.
- **Office Automation System (OAS):** An information system that supports office-related business activities to improve workflow and coordination among employees.

New Technologies

New technologies are being integrated into traditional information systems, including:

- **E-commerce:** The use of the Web to perform business activities and transactions.
- **Enterprise Resource Planning (ERP):** Integrates various information systems within an organization into a single unified platform.
- **Mobile and wireless devices, including mobile commerce (m-commerce):** Enable business operations anytime, anywhere.
- **Open-source software:** Encourages collaboration, flexibility, and innovation in system development.
- **Artificial Intelligence (AI) tools:** Enhance automation, data analysis, and decision-making capabilities.



Systems Analysis and Design

- Identifying **problems**, **opportunities**, and **objectives**.
- Analyzing **information flows** within organizations.
- Designing **computer-based information systems** to solve organizational problems.

A Simple Systems Development Process

- Most organizational systems development processes follow a **problem-solving approach**, which includes the following steps:
  - **Identify the problem**
  - **Analyze and understand the problem**
  - **Define the requirements for the solution**
  - **Identify alternative solutions and select the "best" one**
  - **Design the chosen solution**
  - **Implement the chosen solution**
  - **Evaluate the results** (if the problem remains unsolved, return to Step 1 or 2)

Simplified System Development Process

Simplified System Development Stages	General Problem-Solving Steps
System Initiation	1. Identify the problem. (At the same time, plan a solution approach to the problem.)
System Analysis	1. Analyze and understand the problem.2. Define the requirements for the solution.
System Design	1. Identify alternative solutions and select the “best” one.2. Design the chosen solution.
System Implementation	1. Implement the chosen solution.2. Evaluate the results. (If the problem is not resolved, return to Step 1 or 2.)

What is Information Systems Analysis and Design (ISAD)

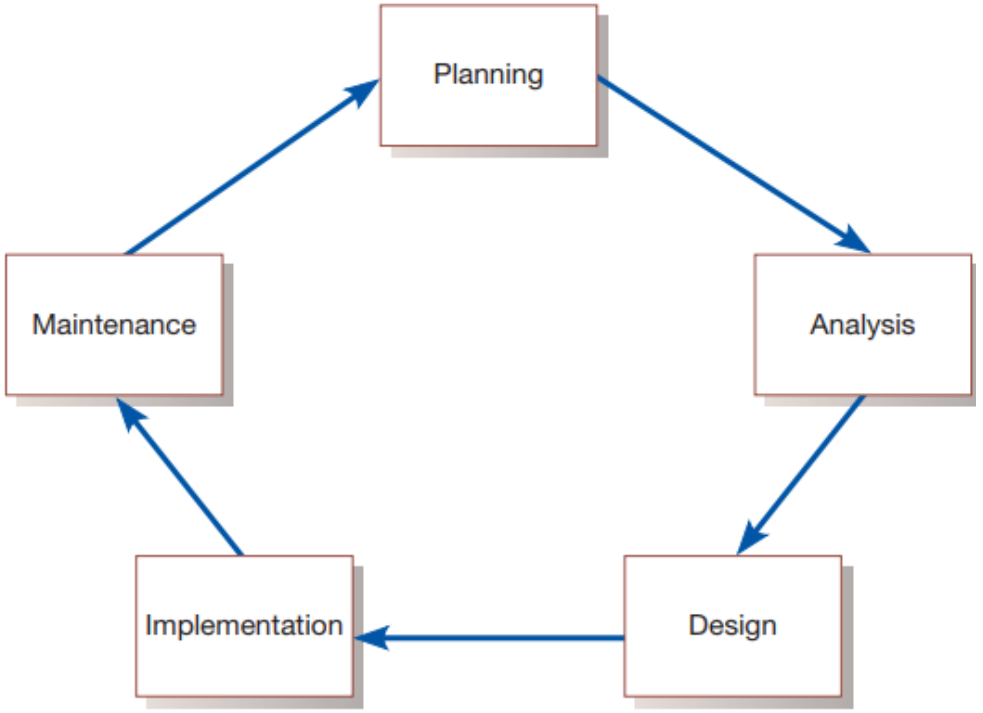
- Information Systems Analysis and Design (ISAD) is defined as a **complex, challenging, and stimulating organizational process** used by a team of business and systems professionals to **develop and maintain information systems**.

What is Information Systems Analysis and Design (ISAD)

- **Objective of ISAD:**
  The core goal is to determine how **people, methods, and information technology** can best be combined to bring about **improvements within an organization**. In essence, ISAD is an **organizational improvement process**.
- **Product:**
  An important outcome of systems analysis and design is **application software**, which is designed to support a **specific organizational function or process** (e.g., inventory management, payroll processing).

Systems Development Life Cycle (SDLC): An Overview

- **Definition:**
  - SDLC is a structured approach to system development.
  - serves as the **fundamental framework** for understanding and organizing the system development process.
- **Phases:**
  - 1.Planning → 2. Analysis → 3. Design → 4. Implementation → 5. Maintenance.



Systems Development Life Cycle (SDLC): An Overview

- **Deliverables:** project plan, requirements, models, tested system, operational support.
- **Example:** University Online Course Registration System lifecycle.
- **Reference:** Hoffer, Ch. 2; Shelly & Rosenblatt, Ch. 1.

SDLC as an Organizational Tool

- The **SDLC** is the **traditional methodology** used to **develop, maintain, and replace** information systems.
- In this course, the SDLC provides a **conceptual and systematic structure** to help students understand the **entire system development process**.

Structure of the SDLC – Five-Phase Model

- The SDLC consists of **five main phases** arranged in a **cyclical, iterative process**:
  1. **Planning:** Identify the need for a new or improved system and develop a detailed **Baseline Project Plan (BPP)**.
  2. **Analysis:** Study current business processes and existing information systems, including **requirements determination** and **requirements structuring**.
  3. **Design:** Transform the proposed solution into detailed **logical and physical system specifications**.
  4. **Implementation:** Include **coding, testing, installation, documentation, training, and user support**.
  5. **Maintenance:** The final phase, where the system is **systematically repaired, updated, and enhanced** to adapt to new business needs.

Evolution of the SDLC

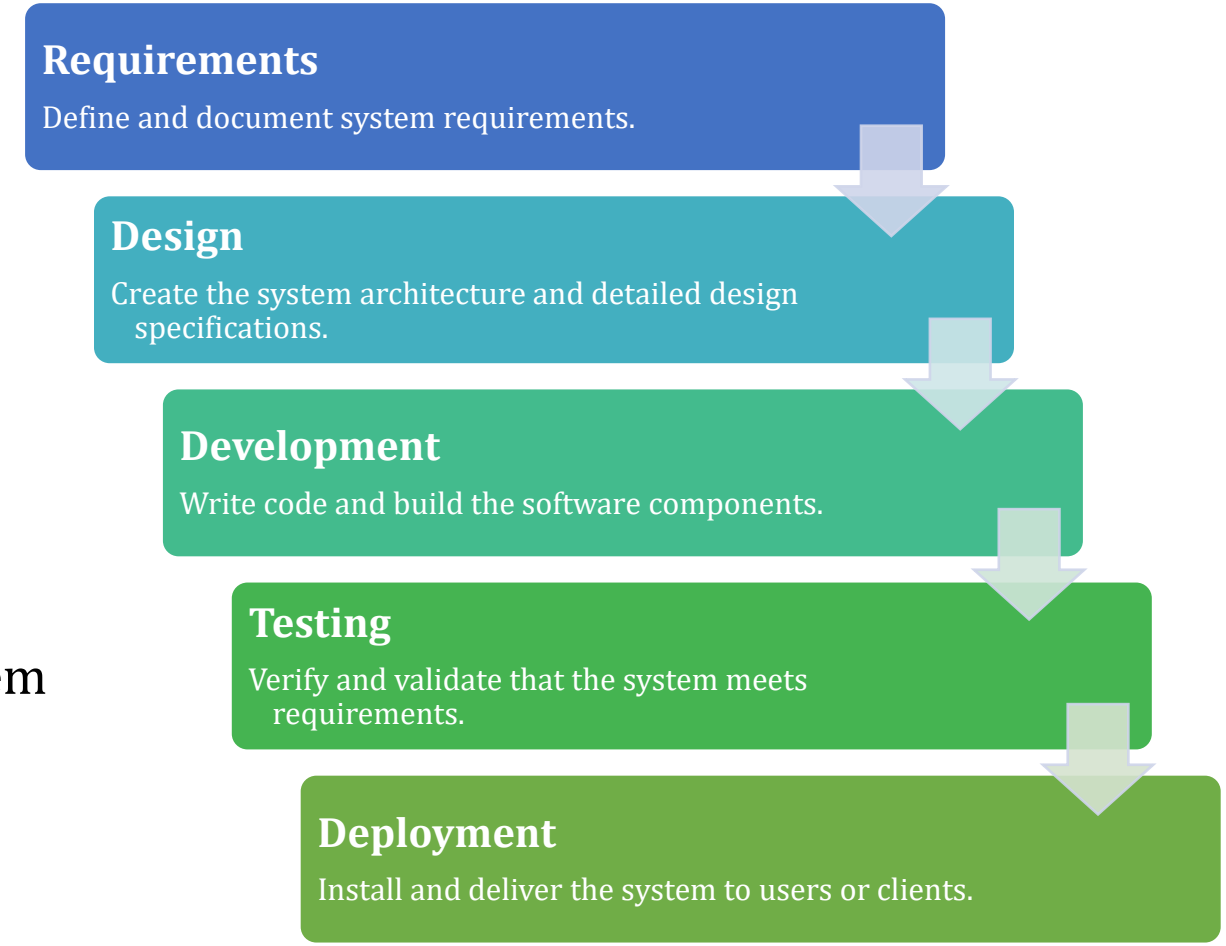
- Although the **traditional SDLC** is often illustrated as a **waterfall model**, in practice it is an **iterative and cyclical process.**
  It provides a **convenient and structured way** to understand all activities involved in **developing and managing information systems**, and it serves as the **organizational foundation** for this course.

Historical Evolution

- The field of **systems analysis and design** began in the **1950s**, when development focused primarily on **processes** due to limited computing power and high costs.
- During the **1960s**, systems development was viewed as an **art**, but by the **1970s**, it became **more disciplined**, resembling **engineering practices**.
- The introduction of **database management systems** using **hierarchical** and **network models** brought structure to data storage and retrieval, gradually shifting the focus from **process-oriented** to **data-oriented** development.

Traditional SDLC Models: The Waterfall Model

- **Features:** Linear, sequential, each phase must be completed before the next.
- **Advantages:**
  - Clear documentation.
  - Easy to manage small projects.
- **Limitations:**
  - Inflexible to change.
  - Late discovery of errors.
- **Example:** Government tax reporting system built with waterfall.
- **Reference:** Pressman, Ch. 2; Hoffer, Ch. 2.



Waterfall Model

The Rise of Agile Development

- The rapid growth of **databases**, **data-driven architectures**, and the **Internet**, along with the emergence of **agile methodologies**, has significantly transformed systems development.
- Modern approaches emphasize **rapid and continuous software delivery**, managed by **small, cross-functional teams** of skilled developers

Agile Methodology

- Introduced through the **Agile Manifesto (2001)**, agile methodologies promote:
  - **Adaptive rather than predictive** planning,
  - **People over roles**, and
  - **Self-adaptive processes** that evolve during development
- **Popular methods:** Scrum, Extreme Programming (XP), Kanban.
- **Example:** Mobile ride-sharing app developed using Scrum sprints.
- **Reference:** Hoffer, Ch. 17; Pressman, Ch. 4.

Key Agile Frameworks

- **eXtreme Programming (XP)**
  - Developed by **Kent Beck and Cynthia Andres (2004).**
  - Characterized by **short cycles**, **incremental planning**, and **continuous testing.**
  - Combines **analysis, design, coding, and testing** into a **single iterative phase.**
  - Uses **pair programming** to improve communication, code quality, and productivity
- **Scrum**
  - Created by **Jeff Sutherland and Ken Schwaber (1995).**
  - The most widely used agile framework (≈ 58 % of companies).
  - Built around **Scrum teams**, **roles (Product Owner, Scrum Master, Development Team)**, and **events** such as **Sprints** (2–4 weeks).
  - Each Sprint delivers a **working product increment**, reviewed and improved through **Sprint Review** and **Sprint Retrospective** meetings

Object-Oriented Analysis and Design (OOAD)

- **OOAD** is recognized as the **standard approach** to modern systems development.
- It combines **data** and **processes (methods)** into **objects**—entities representing real-world elements such as customers or contracts.
- OOAD supports **reuse**, **quality improvement**, and **developer productivity**.
- Importantly, the object-oriented approach shares the **iterative development philosophy** of **agile methodologies**, reflecting their common emphasis on **incremental improvement and adaptability.**

Project Management - Process Management

- **Project Management** is the activity of **defining, planning, directing, and controlling** a project to develop a system that can be **delivered within the assigned time frame and budget**.
- **Process Management** is the **continuous activity** of identifying, improving, and integrating the use of **methodologies** selected by the organization ("processes") with the goal of maintaining **standards across all system development projects**.

Introduction to Information Systems Project Management

- **Role of Project Manager (PM):**
  - Balance scope, time, cost, quality.
  - Manage risk and communication.
- **PM Skills:** Leadership, negotiation, technical knowledge.
- **Example:** ERP system implementation at a manufacturing company.
- **Reference:** Hoffer, Ch. 3; PMBOK Guide (PMI).

Project Management Phases: Initiation & Planning

- **Initiation:** Define objectives, business case, stakeholders.
- **Planning:** Scope, work breakdown structure (WBS), resource allocation, schedule.
- **Tools:** Work packages, milestones.
- **Example:** Gantt chart for university e-learning portal project.
- **Reference:** Hoffer, Ch. 3; Shelly & Rosenblatt, Ch. 2.

Tools for Project Management: Gantt Charts and Network Diagrams

- **Gantt Charts:** Timeline of tasks, progress tracking.
- **Network Diagrams (PERT/CPM):** Dependencies, critical path analysis.
- **Advantages:** Visualizing sequence, resource allocation.
- **Example:** Gantt chart of hospital appointment scheduling system.
- **Reference:** Hoffer, Ch. 3; Pressman, Ch. 25.

Role of the Systems Analyst

- **Organizational Role**
  - A **systems analyst** is the **organizational role** primarily responsible for the **analysis and design** of information systems.
    They act as a bridge between business users and technical teams, ensuring that system solutions meet business needs.
- **Agent of Change and Innovation**
  - The systems analyst serves as an **agent of change and innovation**, helping organizations **adapt to new technologies** and **improve business processes**.
    They promote continuous improvement and digital transformation.

Role of the Systems Analyst

- **Responsibilities**
  - Study organizational problems and needs to determine how **people, methods, and information technology** can be best integrated for improvement.
  - Assist users and managers in identifying **requirements for new or enhanced information services**.
  - Recommend solutions that align with business goals and system feasibility.

Role of the Systems Analyst

- **Key Skills**
  - System documentation and modeling
  - Project management and planning
  - Team coordination and communication

Methodologies, Tools and Techniques

- Modern systems analysis identifies three fundamental components that work together to form the overall **approach to systems development**: **Methodologies**, **Techniques**, and **Tools**.

Methodologies, Tools and Techniques

- **Methodologies**
  - **Definition:** Comprehensive, multi-step approaches that guide the process of **systems development** and influence the **quality of the final information system**.
  - Each organization chooses a methodology consistent with its management style.
  - Most methodologies incorporate several supporting techniques.
  - Examples:
    - Waterfall Model
    - Agile / Scrum Methodology
    - Structured Systems Analysis and Design Method (SSADM)
    - Object-Oriented Analysis and Design (OOAD)

Methodologies, Tools and Techniques

- **Techniques**
  - **Definition:** Specific procedures followed by the analyst to ensure that work is **thorough, structured, and understandable** to others on the project team.
  - Techniques support a wide range of activities, including:
    - Gathering information for requirements
    - Planning and managing project activities
    - Diagramming system logic
    - Designing interfaces, reports, and outputs
  - Examples:
    - Data Flow Diagram (DFD) – for modeling data movement and processes.
    - Entity–Relationship Diagram (ERD) – for modeling data structures.
    - Use Case Modeling – for describing system behavior from the user's perspective.
    - Decision Tables / Decision Trees – for representing business rules.

Methodologies, Tools and Techniques

- **Tools**
  - **Definition:** Typically computer-based programs that make it easier to apply techniques and to follow the principles of a development methodology.
  - Tools help ensure **consistency, efficiency, and accuracy** in system design and documentation.
  - Examples:
    - Microsoft Visio, Lucidchart, Draw.io → for drawing DFDs and ERDs.
    - Rational Rose, StarUML, Visual Paradigm → for UML modeling.
    - JIRA, Trello → for managing Agile or Scrum projects.

Software Development Life Cycle

Software Development Life Cycle

- **Developing and building software is a complex problem:**
  - Software developers **find it very difficult to fully and correctly understand what users need.**
  - User requirements **often change during the development process.**
  - Requirements are **usually described in written documents.**
  - Software developers **often lack expertise in the user's field.**
  - The new software system **must adapt to the user's existing hardware.**

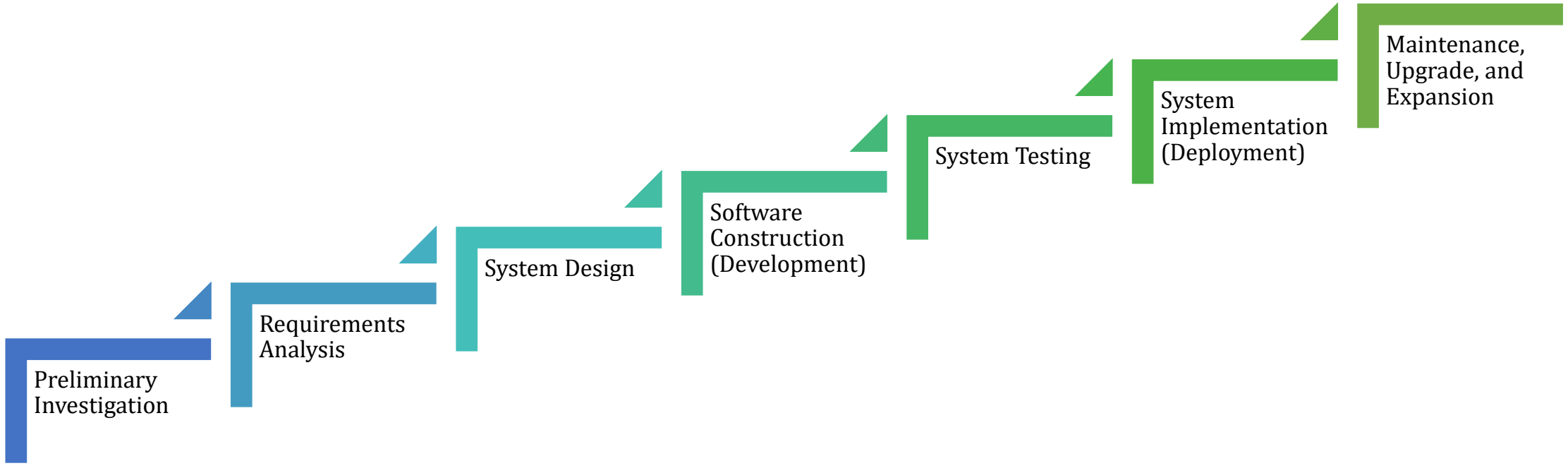
Software Development Life Cycle

- *Common shortcomings in software development:*
  - **Misunderstanding** what users actually need.
  - **Poor adaptability** to changes or new requirements.
  - Software is difficult to maintain, upgrade, and extend.
  - Late detection of bugs and vulnerabilities.
  - Modules interact with each other in an unstable or inefficient way.
  - Team members “step on each other’s toes” or fail to understand one another’s work → wasting time and effort.

Software Development Life Cycle

- *Stages of the software development process:*
  - Preliminary investigation
  - Requirements analysis
  - System design
  - Software construction (development)
  - System testing
  - Implementation and deployment
  - Maintenance, upgrade, and expansion

Software Development Life Cycle



Software Development Life Cycle

1. Preliminary Investigation

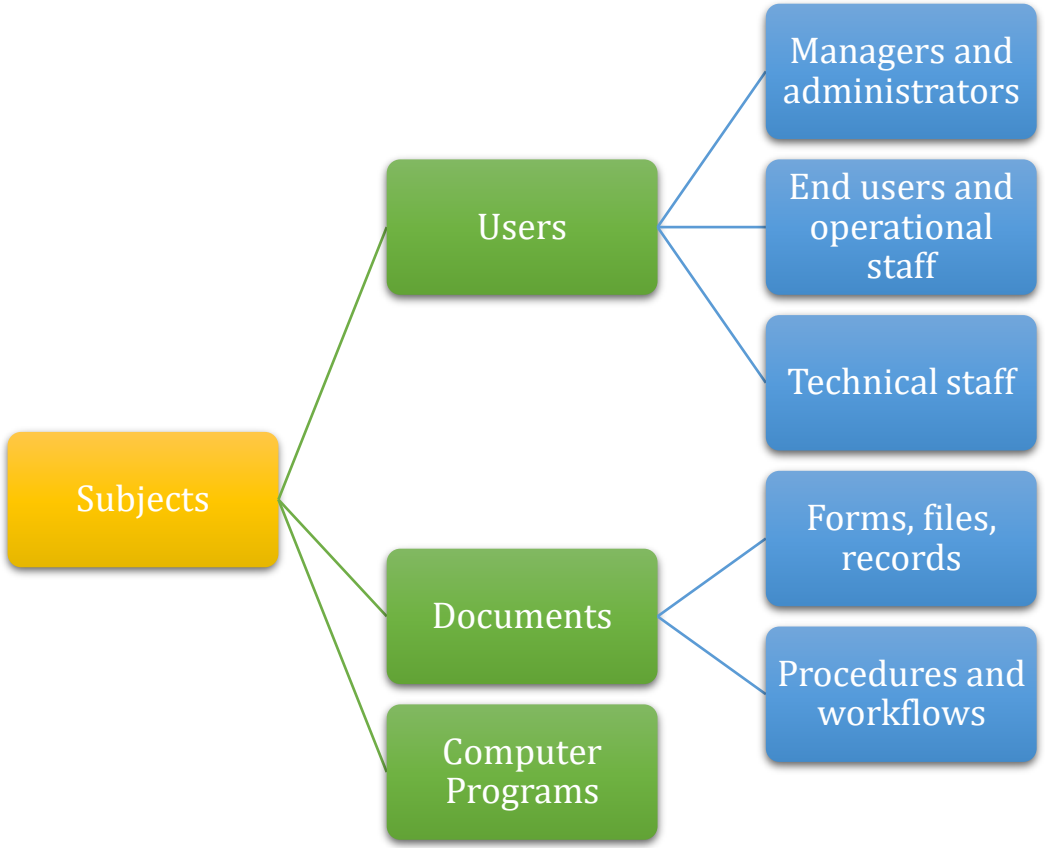
- Review existing operational systems.
- Gather ideas and consult experts in the field.
- Identify external interfaces and collect forms and reports.
- Identify the functions required by users.
- Create prototypes or models of the “future system” to help users visualize it.
- Identify potential risks.

  ➞ Prepare a **feasibility report** for the “future system.”

Software Development Life Cycle

1. Preliminary Investigation

- **Survey Subjects:**

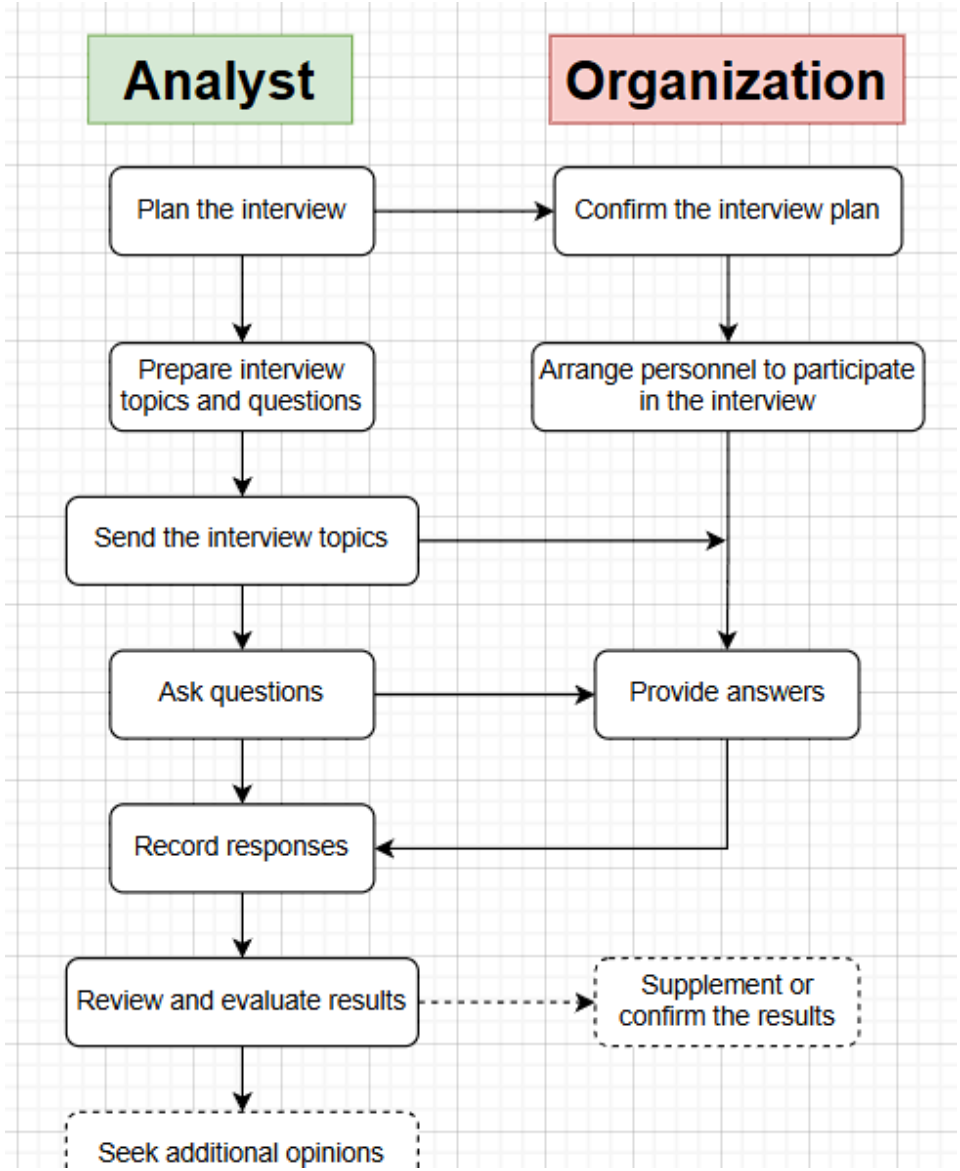


Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

- Interview
- Use questionnaires
- Observation and investigation
- Document study

Interview process



Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

1.1. Interview is a direct dialogue method with relevant individuals to collect information. In this process, the analyst asks questions and the interviewee provides answers.

- **Interview results depend on the following factors:**
  – Preparation for the interview
  – Quality of the questions and note-taking methods
  – Experience and communication skills of the interviewer

Software Development Life Cycle

1. **Preliminary Investigation – Investigation Methods:**

- **Open-ended Questions:**
  - These are questions used to explore or open up issues, requiring respondents to have a certain level of knowledge.
  - The purpose of open-ended questions is to encourage respondents to express all possible opinions within the context of the question.
  - **Example:** "What are your needs when creating the weekly production plan?" or "Do you encounter any difficulties when performing your work?"

Software Development Life Cycle

1. **Preliminary Investigation – Investigation Methods:**

- **Closed-ended Questions:**
  - These are questions where the answer involves selecting one among several predefined options.
  - Closed-ended questions are usually designed in one of the following forms: True/False or Multiple Choice.
- **Example:** "Which of the following is the best feature of the information system you are using?"
  - Easy access to all required data
  - Best system response time
  - Ability to run concurrently with other applications

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

Ending the Interview

- Summarize the key points ⇒ to ensure accurate understanding.
- Review and organize the collected information.
- Prepare an interview report.
- Prepare for future collaboration and leave an open opportunity for both parties.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

Aspect	Open-ended Questions	Closed-ended Questions
Advantages	- Answers are not constrained. - Can generate new ideas.	- Short response time. - Focused and detailed responses, allowing for efficient data collection.
Disadvantages	- Time-consuming. - Difficult to summarize responses. - Responses may go beyond the scope of the question.	- Takes longer to prepare questions. - Useful information may not be included in the predefined options. - Limited in expanding or exploring responses.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

Some notes during the interview process

- Introduce yourself at the beginning of the interview.
- Create a comfortable and friendly atmosphere for the interview.
- Listen attentively, take notes, and avoid giving comments.
- Control the interview: Know how to guide and manage the conversation to avoid digression.
- Use professional language; avoid using technical jargon.
- Do not make the interview too long or prepare too many questions to ask.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

Ending the Interview

- Summarize the key points ⇒ to ensure accurate understanding.
- Review and organize the collected information.
- Prepare an interview report.
- Prepare for future collaboration and leave an open opportunity for both parties.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods

General Interview Plan
System: School Equipment Management
Prepared by: Tran Minh Hoa
Date: March 03, 2025

No.	Topic	Objective	Start Date	End Date
1	Equipment Borrowing and Returning Process	Understand the steps for registering, approving, and recording the borrowing–returning of equipment by lecturers and students.	11/03/25	11/03/25
2	Equipment Inventory Management	Identify procedures for adding new equipment, tracking inventory, conducting audits, and disposing of old items.	12/03/25	12/03/25
3	User Access Control	Understand how user roles and permissions are assigned to staff, technicians, and general users.	13/03/25	13/03/25
4	Hardware and Software Infrastructure	Survey the school's existing server systems, local network, and management software.	14/03/25	14/03/25
5	Reporting and Statistics	Identify the required reports such as equipment by department, usage status, and maintenance schedule.	15/03/25	15/03/25

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods

INTERVIEW SESSION GUIDELINE PLAN

- **System:** School Equipment Management
- **Analyst:** Nguyen Minh Hoa
- **Contact Information:** (IT Office – Tel: 0909 123 456)
- **Interviewee:** Tran Thi Lan
- **Position:** Equipment Management Staff – Administration Office
- **Method:** Face-to-face interview at the office
- **Objectives:**
  - **Data to be collected:** Process of borrowing–returning equipment, forms and records used.
  - **Agreements needed:** Time for providing documents and confirmation of contact person.
- **Appointment Time:**
  - **Start Time:** 09:00 AM – March 16, 2025
  - **End Time:** 10:00 AM – March 16, 2025

Interview Details	
Content	Estimated Time (minutes)
Introduction	2
System Overview	3
Topic 1: Borrowing and Returning Process (Questions & Answers)	10
Topic 2: Inventory and Maintenance Management	12
Topic 3: Forms and Reports Currently Used	8
Summary of Key Points	5
Interviewee's Comments or Questions	10
Conclusion and Next Meeting Agreement	10
Total Estimated Time	60 minutes

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods

- **Interview Result Record Table**
- **Interviewee:** Nguyen Thi Mai
- **Date:** March 12, 2025

Question	Notes
Question 1: How is the borrowing process for equipment conducted in your department?	**Answer**: Lecturers fill out a borrowing form and send it via email or directly to the equipment manager. **Observation**: The process is quite manual and prone to data loss.
Question 2: How are broken devices reported and repaired?	**Answer**: Users record it in a logbook and verbally notify the manager. **Observation**: Information is not updated promptly, easily overlooked.
Question 3: What improvements do you expect from the new system?	**Answer**: The system should track equipment status and send automatic maintenance alerts. **Observation**: High expectations, cooperative attitude.

Software Development Life Cycle

1.2. Using Questionnaires:

Ask respondents to fill in their answers in the provided spaces, then collect and analyze the results.

- A questionnaire usually consists of three parts:
  - **Header section**
  - **Question section:** contains questions arranged by group or by topic area.
  - **Notes section:** includes explanations or clarifications about specific issues in the questions or other comments.

Software Development Life Cycle

1.2. Using Questionnaires:

- Classify questions into groups.
- Classify respondents into groups using the following methods:
  - Key or active participants.
  - Random selection.
  - Purposive selection: meeting certain criteria (e.g., over 2 years of experience, frequent system users, etc.).
  - Classification by type: users, managers, etc.

Comparison of Interview Method and Questionnaire Method

Criteria	Interview	Questionnaire
Richness of Information	High (through multiple channels: responses, observations, gestures, attitudes, etc.)	Medium to low (answers only)
Time	Can be lengthy	Short to moderate
Cost	Can be high	Moderate
Opportunity for Understanding and Discovery	Good: questions can be refined or added by either the interviewer or interviewee	Limited: only after collecting raw data
Confidentiality	Participants know each other	Respondents remain anonymous
Level of Participation	Interviewee plays an important and influential role in the outcome	Passive response; uncertain influence on results

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

1.3. Direct Observation:

- **Advantages:**
  - Ensures the accuracy and reliability of information.
  - Collects comprehensive descriptive data about the system.
- **Disadvantages:**
  - Time-consuming process.
  - May make users uncomfortable while working, as they may feel they are being watched.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:

1.4. Document Study:

- This method helps obtain important information, especially legal or regulatory data.
- In practice, this research often reveals inaccuracies or inconsistencies within the system documentation.

Software Development Life Cycle

1. Preliminary Investigation – Investigation Methods:
1.4. Document Study:

- **Complete documents:**
  - **Transactional documents:** vouchers, letters, notifications, etc.
  - **Archived documents:** books, files, reports, etc.
  - **Summary documents:** reports, statistics, plans.
  - **Organizational and policy documents:** organizational structures, job descriptions, procedures, internal regulations, policies, guidelines, and unwritten rules.
- **Working documents:**
  - **Supplementary documents:** questionnaires, survey forms, etc.
  - **Research documents:** research reports, analyses, etc.
  - **Preparatory documents:** meeting notes, computers, etc.

Comparison of Direct Observation Method and Document Study Method

Factors	Direct Observation	Document Study
Information diversity	High (multiple information channels)	Low (passive and outdated)
Time requirement	Can be long	Short or moderate
Cost	Can be high	Low or moderate
Sustainability and development conditions	Good	Limited: information can only be collected if the original document author is willing to provide it
Reliability	The observed person may change behavior when being monitored	Depends on the nature of the documents; not easily alterable
Related subjects	Interviewed individuals may or may not be relevant, and their engagement depends on whether they know they are being observed	None; the transfer of information is unclear
Key issues	Limited in number and frequency (e.g., screenshots)	Potential usefulness depends on whether documents are updated, as they are often created for other purposes

Software Development Life Cycle

2. Requirement Analysis

- **Main tasks in this phase:**
  - Identify the operating principles of the “future system.”
    - Determine what the system needs to do.
    - Study in detail the required functions and related factors.
  - Work closely with users to clearly define system requirements.
  - Develop various models and diagrams to simulate external impacts on the system.
  - Consult experts for evaluation, verification, and feedback.
- ➡️ Output: **Requirement Specification Report**.

Software Development Life Cycle

2. Requirement Analysis

- This is the most important phase in the software development life cycle.
  - **What should the system do?**
    - …
    - …
    - …
    - …
    - …

Software Development Life Cycle

3. System Design

- **Main tasks in this phase:**
  - Develop a **database structure** suitable for the specifications of each entity.
  - Estimate procedures and describe the **process flow from input to output**.
  - Design **user interfaces** for input/output operations of database entities.
  - Design **user interfaces** for system functions.
  - Create **user-required reports** and **related system reports**.
- ➡️ **Output: System Design Specification Report.**

Software Development Life Cycle

3. System Design

- How does the system meet the requirements stated in the **Requirement Specification Report?**
  - …
  - …
  - …
  - …
  - …

Software Development Life Cycle

4. Software Construction (Implementation)

This is the stage where actual coding takes place.

- **Main tasks during this phase:**
  - Developers use the **Design Specification** to implement the **database**.
  - Tasks may be divided among developers to reduce development time.
  - **Unit testing:** Each developer tests their own module.
  - **Independent testing:** Other team members test modules written by someone else.
  - **Integrate multiple independent modules together.**
- ➡️ **The software system becomes operational.**

Software Development Life Cycle

5. System Testing

- **Main tasks in the system testing phase:**
  - Test the system based on the details in the **Requirement Specification.**
  - Act as **end users** to verify system functionality.
  - Consult **domain experts** to evaluate the system's performance.
- ➡️ **THE SOFTWARE SYSTEM IS COMPLETE.**

Software Development Life Cycle

6. Implementation and Deployment

- **Main tasks in this phase:**
  - Install the **server** and **database system**.
  - Install software on the users' personal computers as required.
  - Create **backups** for both the database and the software.
  - Provide **user training** on how to operate the software.
  - Perform **regular database backups** for recovery in case of failure.
  - Have users **sign the software system acceptance report**.
- ➡️ **SOFTWARE SYSTEM ACCEPTANCE REPORT.**

Software Development Life Cycle

7. Maintenance, Upgrade, and Expansion

- *When there are requests from users*
- **Main tasks in this phase:**
  - Survey and analyze user requests.
  - Perform maintenance, upgrades, and system expansion.
  - Deliver the updated system after maintenance, upgrade, or expansion.
- ➡️ **REPORT ON MAINTENANCE, UPGRADE, AND EXPANSION ACTIVITIES.**

Function-Oriented and Object-Oriented Approaches

Function-Oriented and Object-Oriented Approaches

- **Function-Oriented Approach**
  - This is the **traditional approach** in software engineering:
    - Ask users what information and functions they need.
    - Design a **database (DB)** that stores that information.
    - Build **forms** and tools for users to input data, print reports, and perform the required functions.
  - ➡️ **Focuses mainly on data and basic functions derived from user requirements.**

Function-Oriented and Object-Oriented Approaches

- **Function-Oriented Approach**
  - **Difficulties:**
    - **Data and functions** are tightly coupled; when the **database (DB)** changes, one or more **forms** must also be modified.
    - **System expansion or modification** becomes difficult because it requires updating the **DB and forms**.
    - Hard to **adapt or transfer** the application system to another organization.
  - ➡️ **Solution: A new approach.**

Function-Oriented and Object-Oriented Approaches

- **Object-Oriented Approach**
  - This is the **modern approach** in software engineering:
    - Divide the application into multiple objects.
    - Each object operates independently.
    - The system is a combination of objects (Data and Functions).
  - ➡️ **Focuses mainly on the data and functions of each object within the software system.**

Function-Oriented and Object-Oriented Approaches

- **Advantages of the Object-Oriented Approach**
  - **Reusability:**
    - Objects created and used in one system can be reused in other systems.
    - Since they are thoroughly tested in previous systems, this reduces errors and increases development speed.
    - Leads to the creation of software with high adaptability and robustness.

Object-Oriented Method and Software Development Life Cycle

- The steps in the software development life cycle shift the approach toward real-world objects, making it easier to understand for both users and development team members.
- Facilitates modeling to present ideas about data, processes, and the interaction between objects in the system.
- Provides multiple perspectives on objects, enhancing the system's adaptability for deployment across various organizations.

Exercise 1

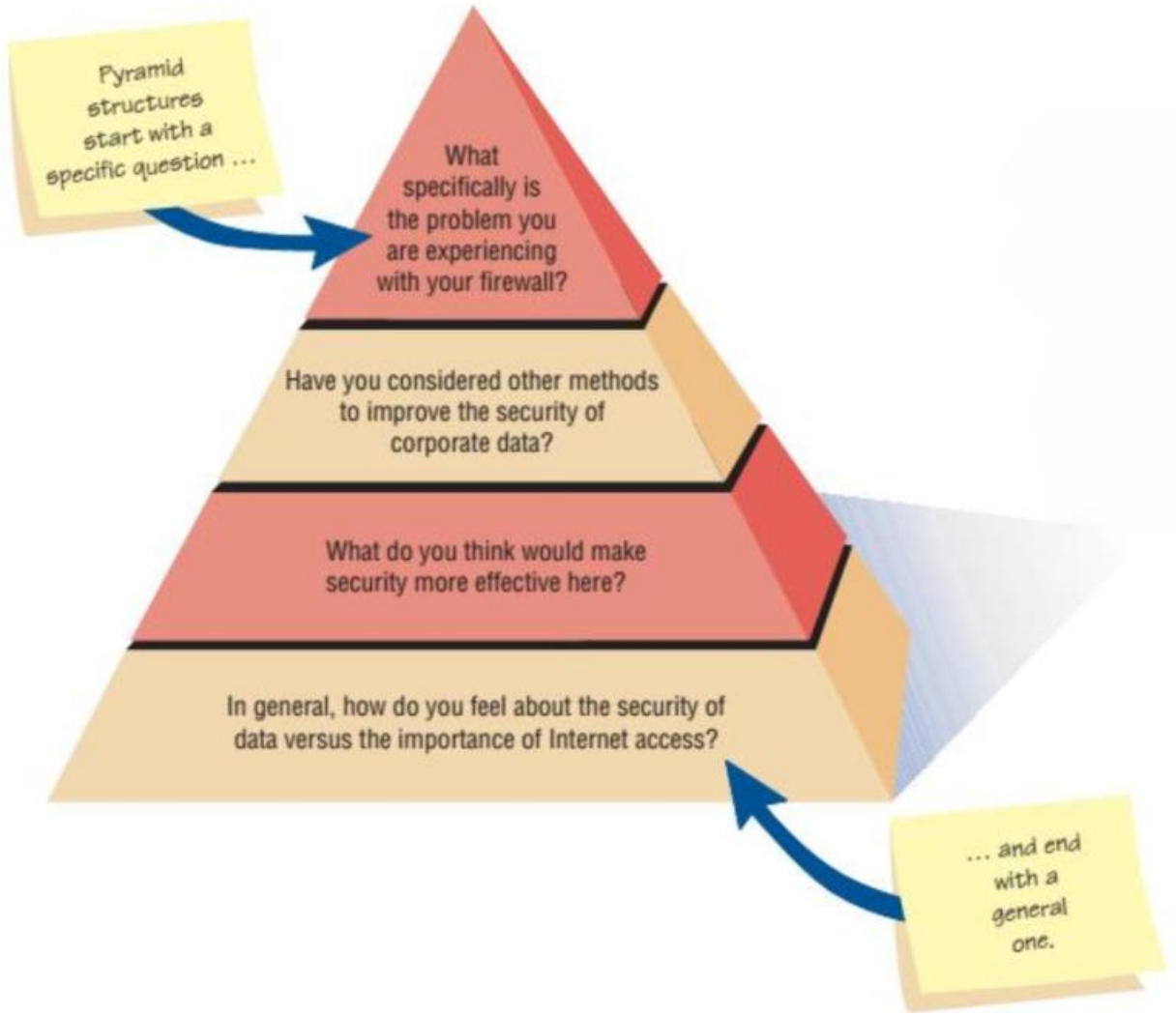
- For the system your group is working on, perform the following tasks when conducting interviews to determine user requirements:
  - Define the **objectives** of the interview.
  - Identify the **tasks** you will perform to **prepare** for the interview.
  - Create **15 (or over) key interview questions** (including both closed and open questions), and **number them from 1 to 15**.
  - **Arrange the questions** according to your **interview strategy** (ordered by question numbers).

Exercise 1

- **Question Sequencing**
  - **Pyramid Structure**
    - Start with **closed-ended questions**, followed by **open-ended questions**
  - **Funnel Structure**
    - Start with **open-ended questions**, followed by **closed-ended questions**
  - **Diamond Structure**
    - Start with **closed-ended questions**, then move to **open-ended questions**, and finish with **closed-ended questions**

Exercise 1

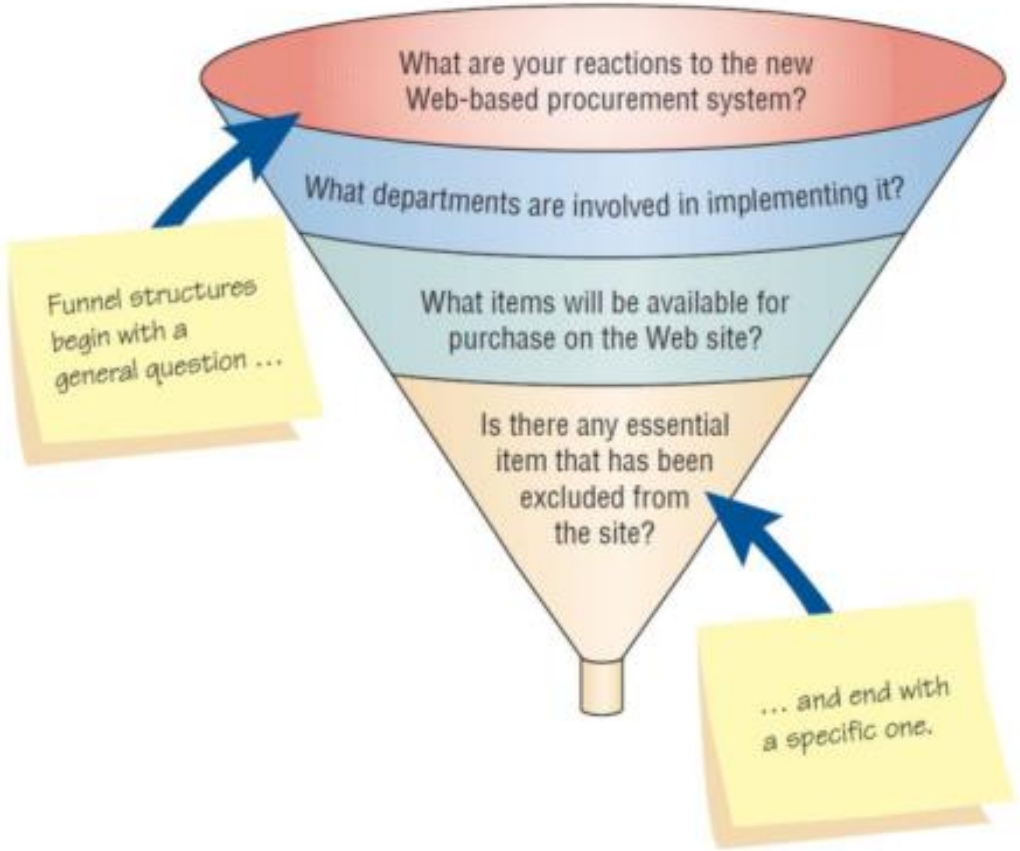
- **Pyramid Structure**
  - Start with very detailed, typically **closed-ended questions**
  - Broaden the discussion by using **open-ended questions** and **more general answers**
  - Useful when the **interviewee needs warming up** to the topic or appears **reluctant** to discuss it



➡️ **Pyramid structure guides interviews from specific questions to more general ones.**

Exercise 1

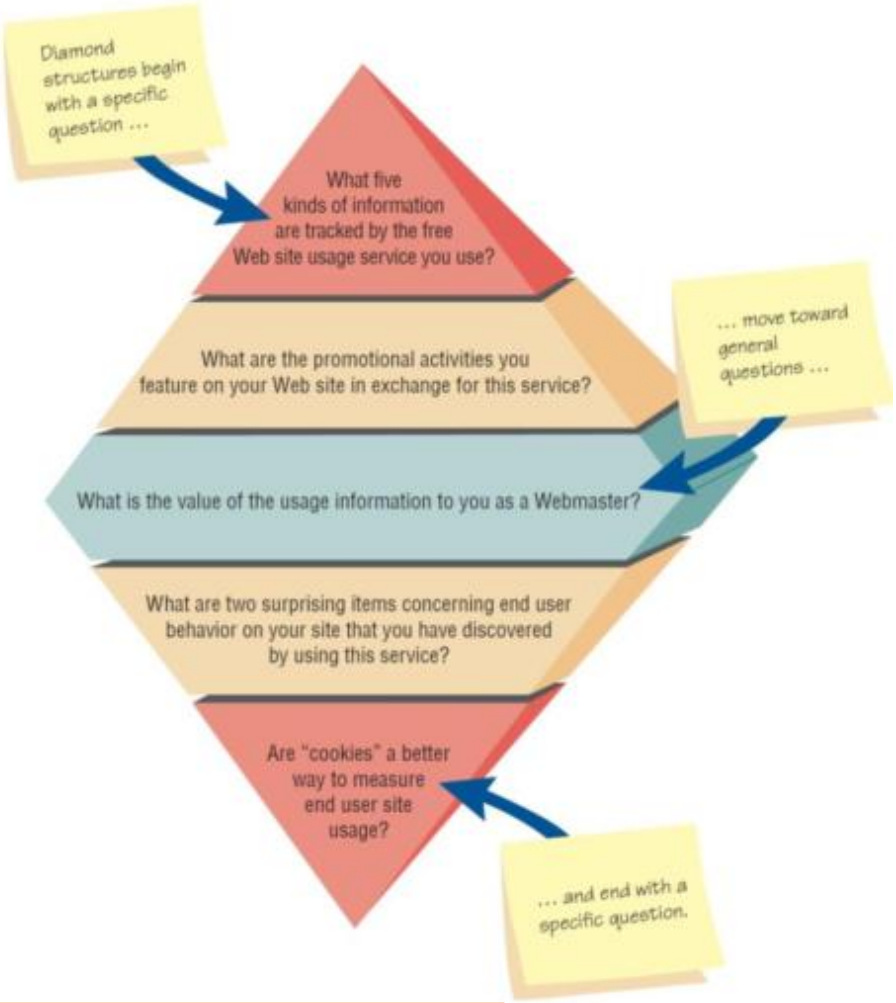
- **Funnel Structure**
  - Start with **general or open-ended questions**
  - Conclude by **narrowing down the responses** possibly using **closed-ended questions**
  - Provides an **easy and non-threatening** way t begin an interview
  - Useful when the **interviewee has emotional involvement** or strong feelings about the topi



The funnel structure allows interviews to begin with broad questions and then narrow down to specific ones.

Exercise 1

- **Diamond Structure**
  - A **diamond structure** begins with a **very specific question**.
  - Then, **broader and more general issues** are explored.
  - It **concludes with a specific question** again.
  - This structure **combines the strengths** of both the **pyramid** and **funnel** approaches.
  - It **takes more time** to conduct than other structures.



The diamond structure allows interviews to combine the benefits of both pyramid and funnel structures.

Exercise 2

- For the system your team is developing, design a **questionnaire** to collect user information for **requirements determination**. Perform the following tasks:
- **Define the objectives of the questionnaire**
  → What kind of information do you want to collect?
  (e.g., user satisfaction, functional needs, current difficulties, etc.)
- **Identify the target respondents**
  → Who will answer your questionnaire?
  (Students, lecturers, administrative staff, end-users, etc.)
- **Prepare the questionnaire content**
  → Design **10–12 questions**, including both **closed-ended** and **open-ended** types.
  → Each question must be **numbered from 1 to 12**.
- **Classify the question types**
  → Indicate the type of each question:
  Multiple-choice, Likert scale, Open-ended, Yes/No, etc.
- **Present the final questionnaire layout**
  → Include a **title**, **brief instructions**, **respondent information** (anonymous or not), and a **thank-you note**.

Exercise 2

- **Questionnaires**
- Questionnaires are useful for collecting information from key members of an organization about:
  - **Attitudes**
  - **Beliefs**
  - **Behaviors**
  - **Characteristics**

Exercise 2

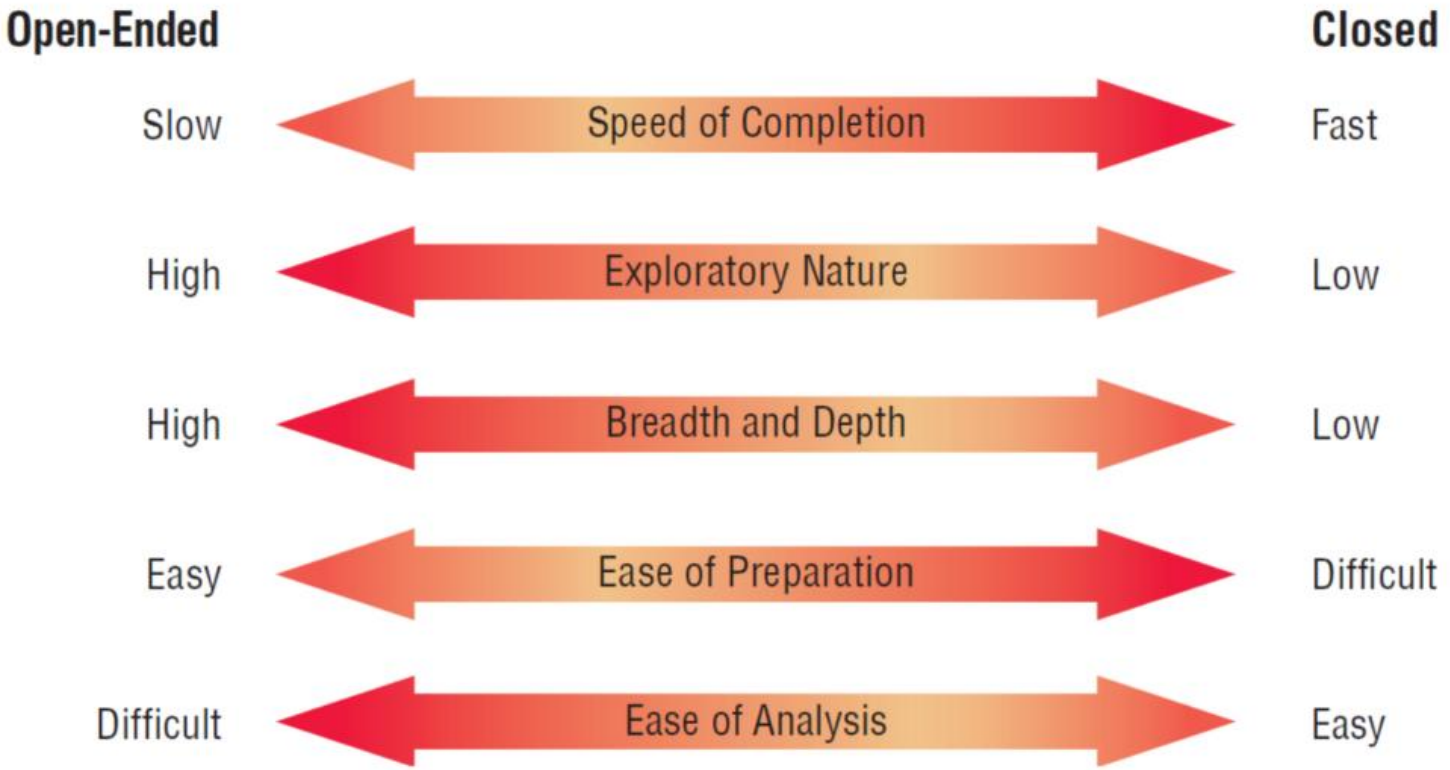
- **Planning for the Use of Questionnaires**
  - Members of the organization are **widely distributed**
  - **Many members** are involved in the project
  - **Preliminary investigation** is necessary
  - It is important to **address issues before conducting interviews**

Exercise 2

- **Types of Questions**
  - Questions are generally designed in one of the following two types:
    - **Open-ended questions**
      - Try to anticipate the kinds of responses you might receive
      - Very suitable for gathering opinions
    - **Closed questions**
      - Used when all possible choices can be listed
      - When choices are **mutually exclusive**

Exercise 2

- Trade-offs Between Using Open-ended and Closed-ended Questions in a Questionnaire



Exercise 2

- **Questionnaire Language**
  - **Simple**
  - **Specific**
  - **Short**
  - **Not patronizing**
  - **Free of bias**
  - **Addressed to those who are knowledgeable**
  - **Technically accurate**
  - **Appropriate for the reading level of the respondent**

Exercise 2

- **Measurement Scales**
- There are two different types of measurement scales:
  - **Nominal scale**
  - **Interval scale**

Exercise 2

- **Nominal Scales**
  - Nominal scales are used to **classify objects**
  - They represent the **weakest form of measurement**
  - The data can be **summarized or counted**

What type of software do you use the most?

1 = Word Processor

2 = Spreadsheet

3 = Database

4 = An Email Program

Exercise 2

- **Interval Scales**
  - Interval scales are used **when equal intervals exist** between measurement points
  - There is **no absolute zero**
  - Examples of interval scales include **Fahrenheit** or **Centigrade (Celsius)** temperature scales

How useful is the support given by the Technical Support Group?				
NOT USEFUL AT ALL				EXTREMELY USEFUL
1	2	3	4	5

Exercise 2

- **Reliability and Validity**
  - **Reliability** refers to the **consistency** of responses — the ability to obtain **similar results** if the same questions are administered again under the same conditions.
  - **Validity** is the extent to which the question **measures what it is intended to measure**.

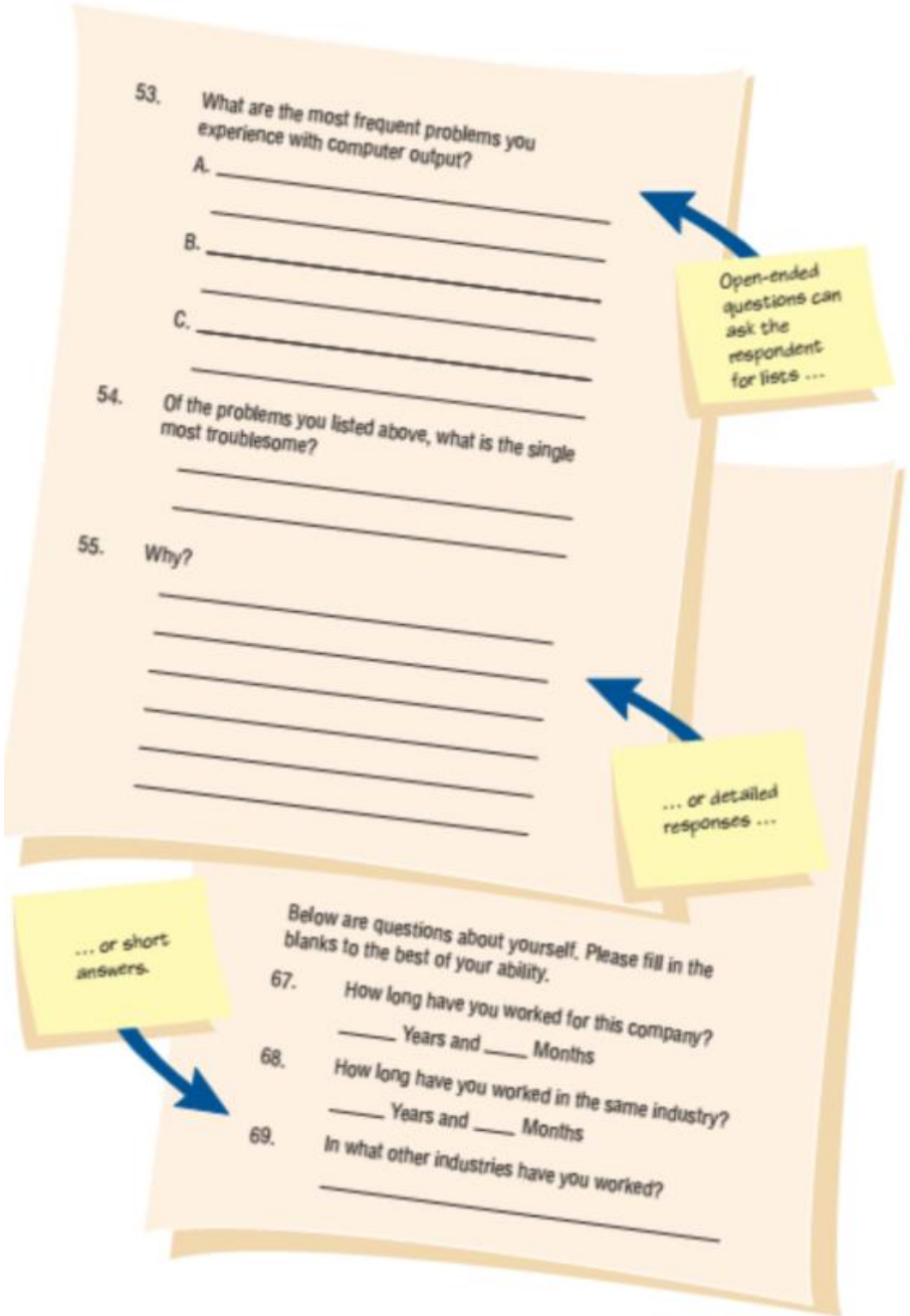
Exercise 2

- **Problems with Measurement Scales**
  - **Leniency**
    - Caused by **raters who are too generous or lenient**
    - The solution is to **shift the "average" categories** slightly to the **left or right of the midpoint**
  - **Central Tendency**
    - Occurs when respondents tend to choose **average or neutral responses** for everything.
      - Improve by **creating smaller differences at the ends** of the scale.
      - **Adjust the intensity or length** of descriptors.
      - **Use a scale with more points** to encourage varied answers.
  - **Halo Effect**
    - Happens when **an impression formed by one question influences the response** to subsequent ones.
    - The solution is to **place each characteristic or set of items on separate pages** to minimize bias.

Exercise 2

- **Questionnaire Design**
  - Provide **adequate spacing**
  - Ensure **enough space** for respondents to **write or type their answers**
  - Make it **easy for respondents to mark their choices**
  - Maintain **consistency in layout and style**

Exercise 2


53. What are the most frequent problems you experience with computer output?
A.
B.
C.
54. Of the problems you listed above, what is the single most troublesome?
55. Why?
Open-ended questions can ask the respondent for lists ...
... or detailed responses ...
... or short answers.
Below are questions about yourself. Please fill in the blanks to the best of your ability.
67. How long have you worked for this company?
_____ Years and _____ Months
68. How long have you worked in the same industry?
_____ Years and _____ Months
69. In what other industries have you worked?

Exercise 2

- **Question Order**
  - Place the **most important questions first**
  - **Group related questions** or sections with similar content together
  - Begin with **non-controversial questions** to make respondents comfortable

Exercise 2

- When designing a **web-based survey**, remember that there are **different ways** to capture responses

Name	Appearance	Purpose
One-line text box		Used to obtain a small amount of text and limit the answer to a few words
Scrolling text box		Used to obtain one or more paragraphs of text
Check box		Used to obtain a yes-no answer (e.g., Do you wish to be included on the mailing list?)
Radio button		Used to obtain a yes-no or true-false answer
Drop-down menu		Used to obtain more consistent results (Respondent is able to choose the appropriate answer from a predetermined list [e.g., a list of state abbreviations])
Push button		Most often used for an action (e.g., a respondent pushes a button marked "Submit" or "Clear")

Exercise 2

- **Methods of Administering Questionnaires**
  - Gather all relevant respondents **together at the same time**
  - Conduct **individual surveys** using questionnaires
  - Allow respondents to **self-administer the questionnaire**
  - **Mail** the questionnaire to respondents
  - Conduct the survey **via the Web or through email**

Exercise 2

- **Administering Questionnaires via Electronic Means**
  - Reduce costs
  - Collect and store results electronically