System Analysis and Design

Presented by

Dr. Amira M. Gaber

Lecture 1

computer application

- ***A computer application*** is a computer software program that executes on a computing device to carry out a specific function or set of related functions. Sometimes, computer application is shortened to app (such as an iPhone app or a Facebook app).

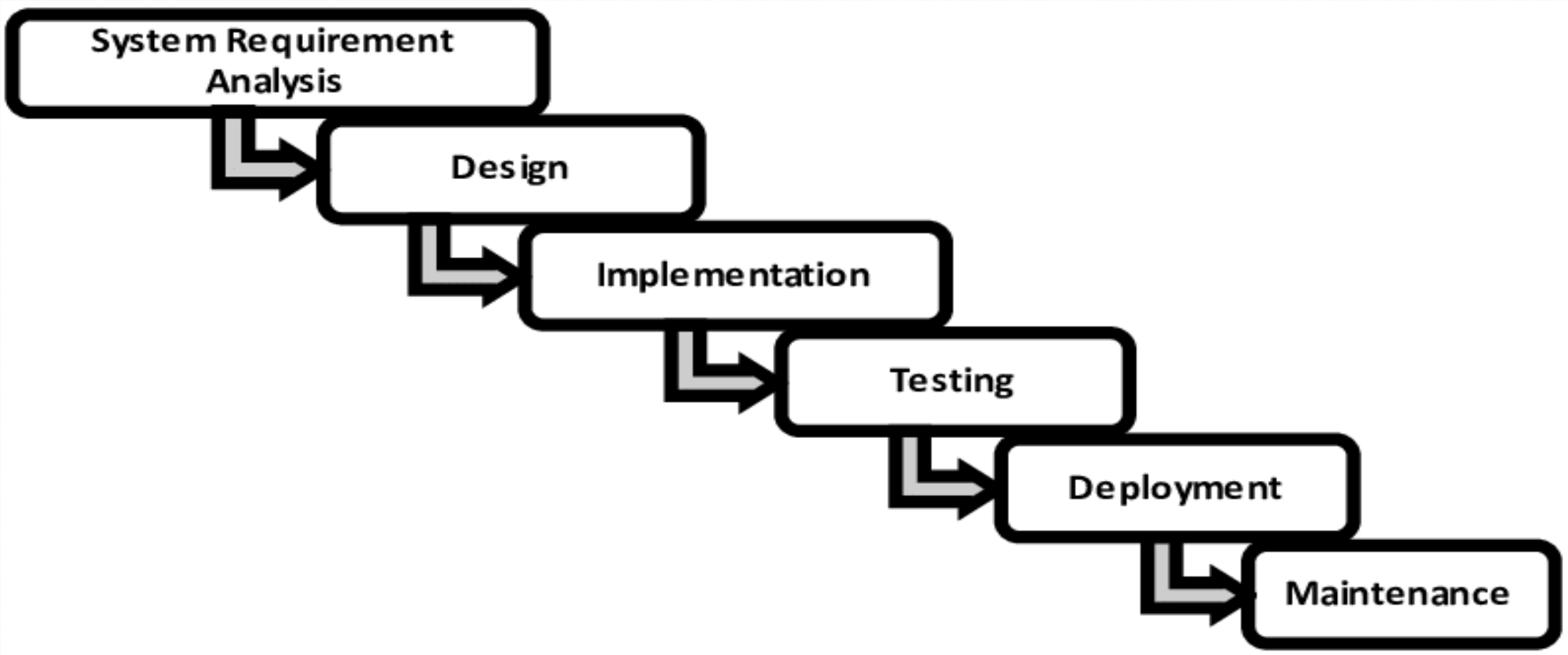
information system

- An ***information system*** is a set of interrelated computer components that collects, processes, stores (usually in a database), and provides as output the information needed to complete business tasks. Although these terms are sometimes used synonymously, an application usually refers to only the computer software involved, whereas an information system may include the software, the database, and even the related manual processes.

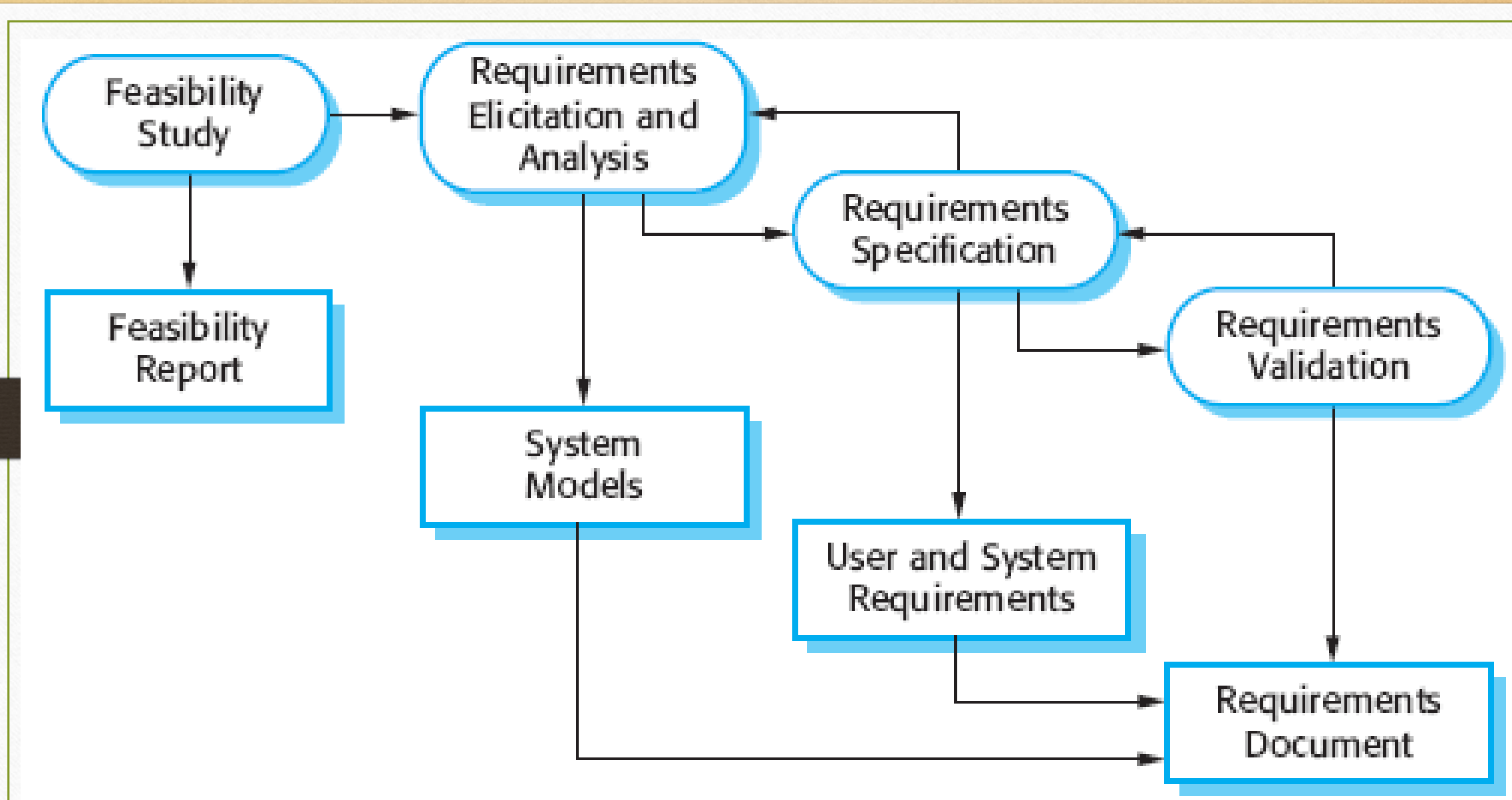
What is the software Development Life Cycle (SDLC)?

- **Software Development Life Cycle (SDLC)** is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.
- SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.
- **The life cycle** defines a methodology for improving the quality of software and the overall development process

Phases of SDLC



System Analysis Phase


Feasibility Study
Requirements Elicitation and Analysis
Requirements Specification
Requirements Validation
Feasibility Report
System Models
User and System Requirements
Requirements Document

System Analysis Phase

- A ***requirement*** is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

- ***Requirements determination*** involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

Types of Requirements

- **Business requirements** describe why the organization is undertaking the project. They state some benefits that the developing organization or its customers expect to receive from the product. Business requirements bring the project owner, stakeholders and the project team on the same objective. we consider the following business requirements.
- Problem Statement
- Project Constraints (Budget, Schedule, and Resources)
- Project Objectives
- Project Scope Statements
- The purpose of the business process analysis is to determine how the business process will work. It is often necessary to resolve deficiencies in the business process before trying to automate it.

Types of Requirements

User requirements: is a planning document that specifies *what* the software or system needs to do. It is written from the point of view of the end user and does not need to be technical or complicated. It is clear, unambiguous, well explained and concise. It helps the systems designer or software engineer fully understand a client's needs, and can be used to plan a timetable, estimate costs.

Types of Requirements

System requirements is a requirement at the system level that describes a function or the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements. System requirements are expressed in an appropriate combination of textual statements, views, and non-functional requirements. System requirements express the levels of safety, security, reliability which will be necessary.

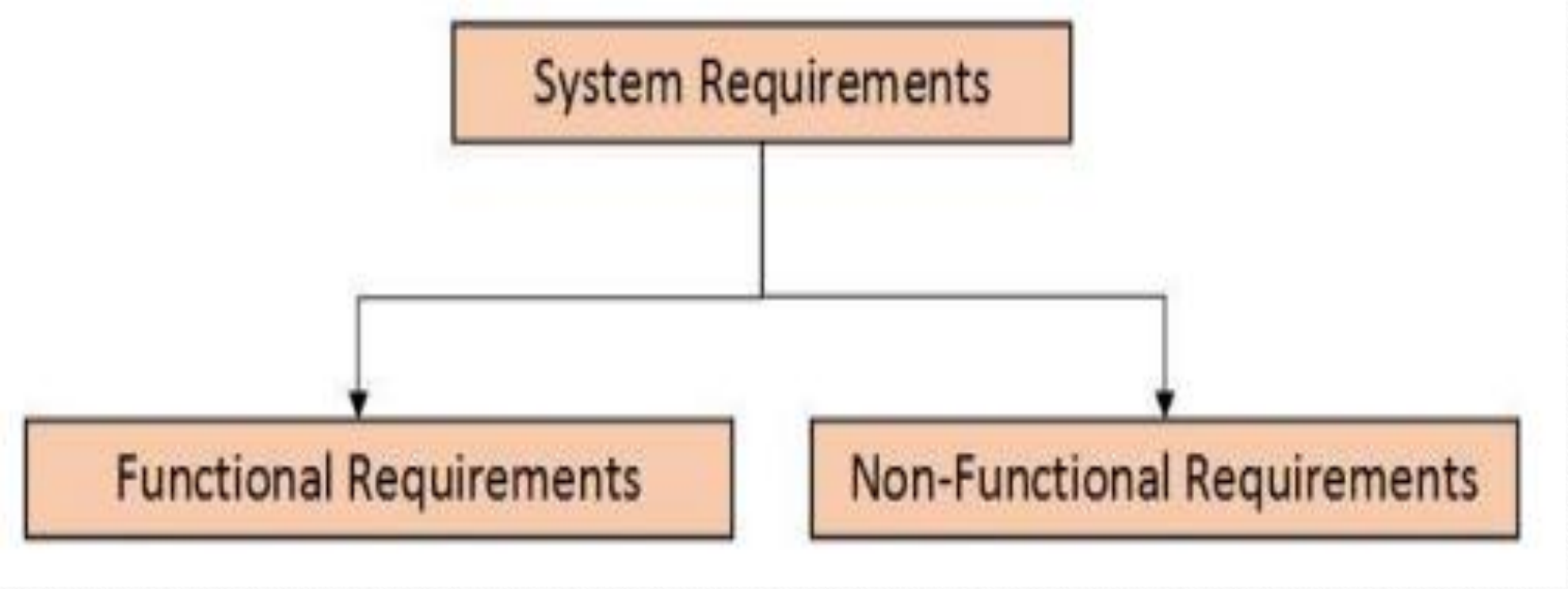
User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

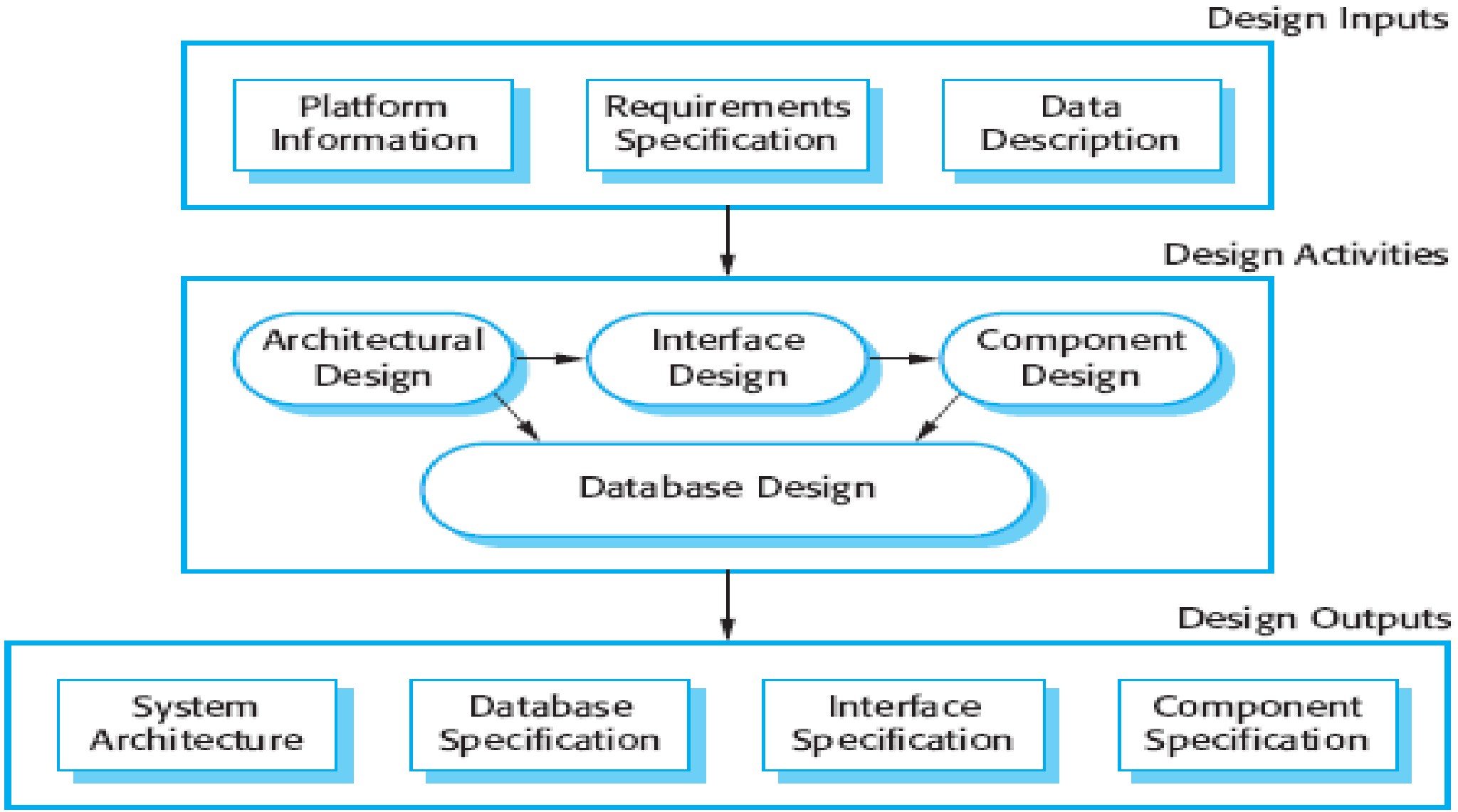
System Requirements



Functional and Non-Functional requirements

- **Functional requirements** are the activities that the system must perform (i.e., the business uses to which the system will be applied). Functional requirements are based on the procedures and rules that the organization uses to run its business. Sometimes, they are well documented and easy to identify and describe.
- **Non-functional requirements** are characteristics or constraints of the system other than those activities it must perform or support. It is not always easy to distinguish functional from nonfunctional requirements.

System Design Phase


Design Inputs
Platform Information
Requirements Specification
Data Description
Design Activities
Architectural Design
Interface Design
Component Design
Database Design
Design Outputs
System Architecture
Database Specification
Interface Specification
Component Specification

System implementation Phase

Phase 3: Software implementation

The implementation stage of software development is the process of converting a system specification into an executable system. It always involves processes of software design and programming but, if an incremental approach to development is used, may also involve refinement of the software specification.

System Testing Phase

Phase 4: Software Testing

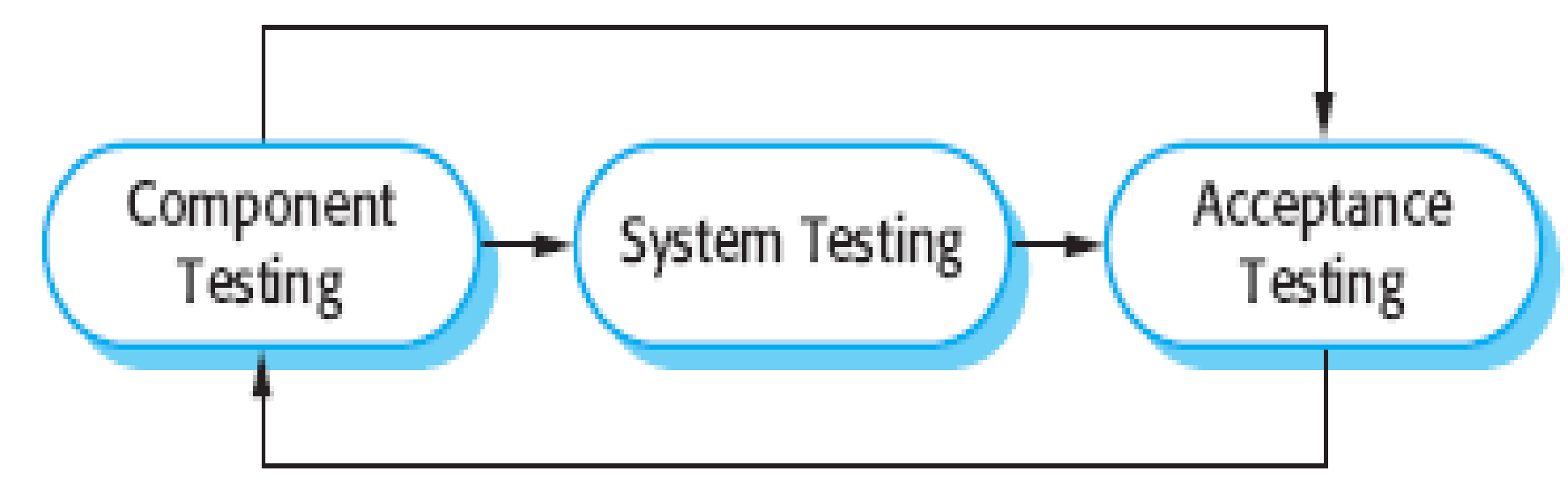
Software Validation is a process of evaluating software product, so as to ensure that the software meets the pre-defined and specified business requirements as well as the end user’s/customers' demands and expectations.

Phase 4: Software Testing(Cont.)

Both, the **verification** and validation is a software testing activity, and verification is followed by the validation. Validation is usually carried out at the end of the software development

three-stage testing process in which system components are tested then the integrated system is tested and, finally, the system is tested with the customer's data.

The stages of the testing process



Phase 4: Software Testing(Cont.)

- ***Component testing*** The components making up the system are tested by the people developing the system. Each component is tested independently, without other system components.

- ***System testing*** System components are integrated to create a complete system. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

Phase 4: Software Testing(Cont.)

- ***Acceptance testing*** This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system customer rather than with simulated test data.
- ***Acceptance testing is sometimes called 'alpha testing'***
- The alpha testing process continues until the system developer and the client agree that the delivered system is an acceptable implementation of the requirements.

Phase 4: Software Testing (Cont.)

When a system is to be marketed as a software product, a testing process called **'beta testing'** is often used. Beta testing involves delivering a system to a number of potential customers who agree to use that system. They report problems to the system developers.

Phase 4: Software Testing (Cont.)

- **Validation** is an actual testing performed on the software product. It gives answer to our query of ***"Are we developing the right software product?"***

validation process includes:

- Gathering and analysis of the specifications and requirements.
- Based on specifications and requirements, preparation of test strategies, plans and cases, that seems fit for use.
- Go for testing the boundary values along with stress and functionalities test.
- Test the error message.
- Conducts software evaluation, as it ensures that the software meets the all pre-decided requirements and is acceptable for use.

Software Validation	Software Verification
1. Validations is a dynamic mechanism of testing a software product.	1. Verification is a static practice of verifying various aspects of a software product.
2. Ensures that the software is in compliance with user requirements.	2. Ensures that the product is developed is as the requirements and design specifications.
3. It involves the execution of the code.	3. It does not involve the execution of the code.
4. It is performed by the testing team.	4. It is performed by the QA team.
5. Involves white box testing, black box testing and gray box testing.	5. It involves reviews, meeting and inspections done by the QA team.
6. This is a high level testing.	6. This is a low level testing.

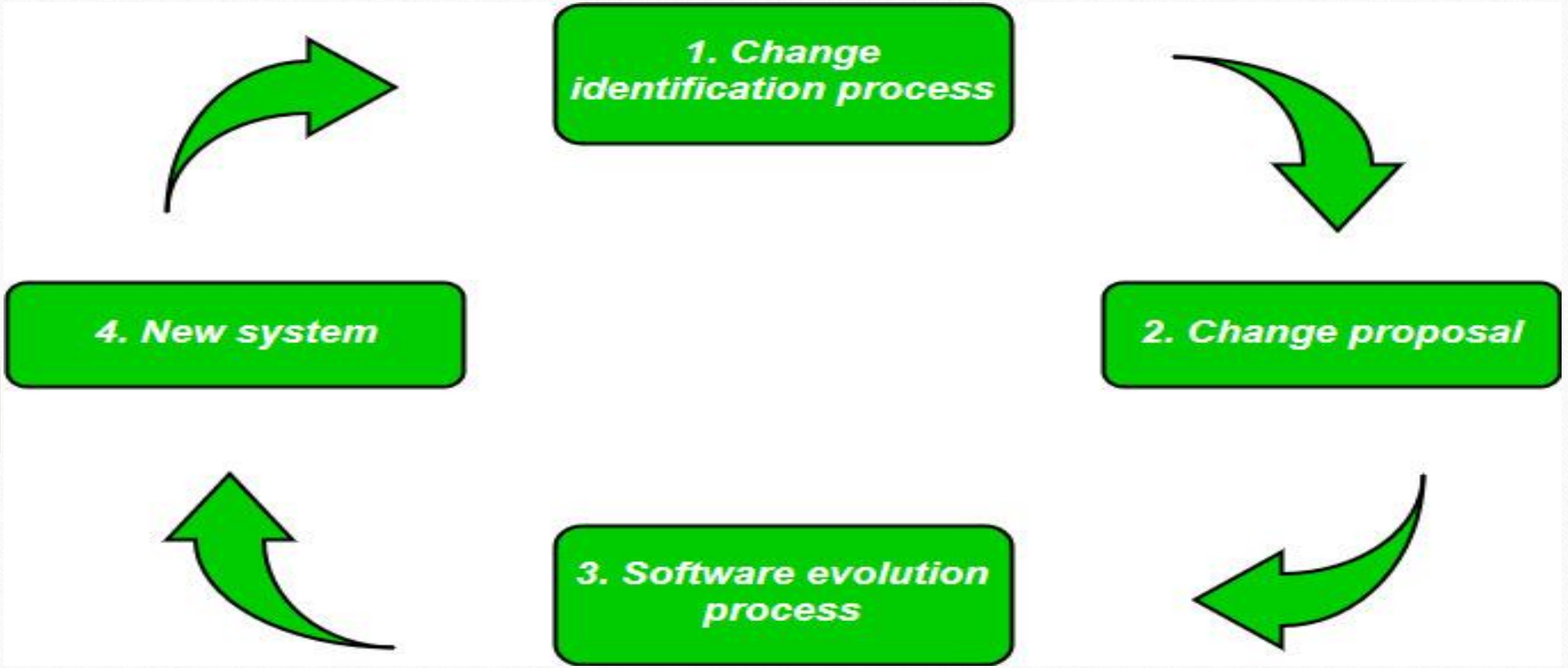
Phase 5: Software Evolution and Maintenance

Software Evolution and Maintenance is referred to as the process of developing, maintaining and updating software for various reasons. Software changes are inevitable because there are many factors that change during the life cycle of a piece of software. Some of these ***factors*** include:

- Requirement changes
- Environment changes
- Errors or security breaches
- New equipment added or removed
- Improvements to the system

Phase 5: Software Evolution and Maintenance (Cont.)

- The evolution process includes fundamental activities of change analysis, release planning, system implementation and releasing a system to customers.
- The cost an impact of these changes are accessed to see how much system is affected by the change and how much it might cost to implement the change. If the proposed changes are accepted, a new release of the software system is planned. During release planning, all the proposed changes (fault repair, adaptation, and new functionality) are considered.



Evolutionary Process

Time for Questions


Thank you!