# Software Engineering

# Software Development Life Cycle

Dr. Sayed AbdelGaber
Professor
Faculty of Computers and Information
Helwan University

# Why Software Engineering?

- More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.

- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

- Much of our software is delivered late, over budget, and with residual faults. Software engineering is an attempt to solve these problems.

# Essential Attributes of Good Software

| Product characteristic | Description |
|---|---|
| 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. |
| Dependability and security | Software dependability includes a range of characteristics including reliability, security and safety. Dependable 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. |

# Software Development Objectives

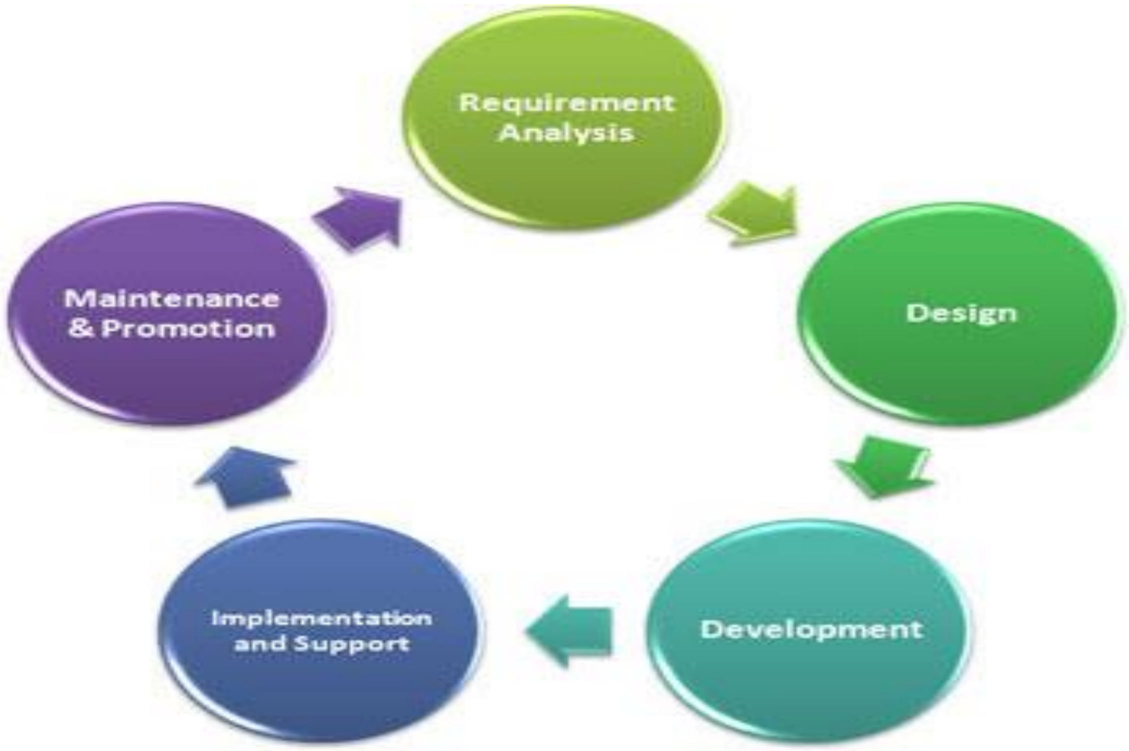
- Meeting user’s needs
- Low cost of production
- High performance
- Portability
- Low cost of maintenance
- High reliability
- Delivery on time

# Categories of software

- System software
- Application software
  - ✓ Games
  - ✓ Information systems
  - ✓ Real-time systems
  - ✓ Embedded systems
  - ✓ Office software
  - ✓ Scientific software

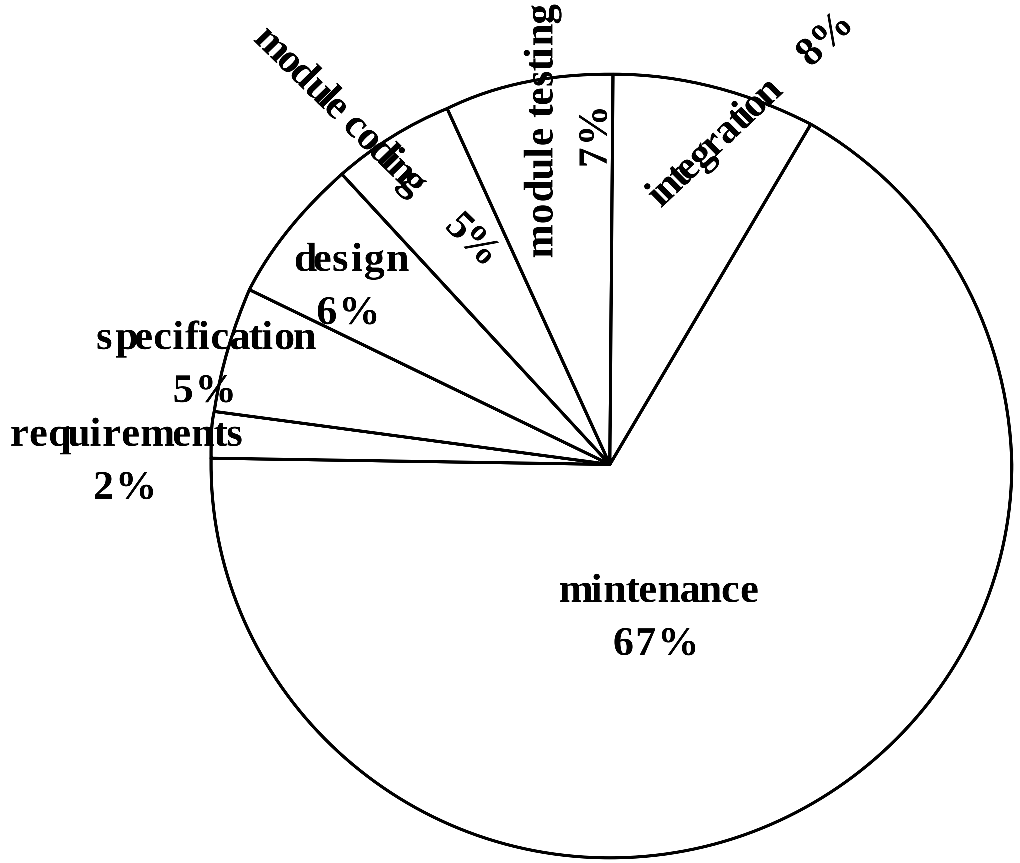
**Software can either be off-the-shelf or tailor-made.**

# Software Development Life Cycle (SDLC)

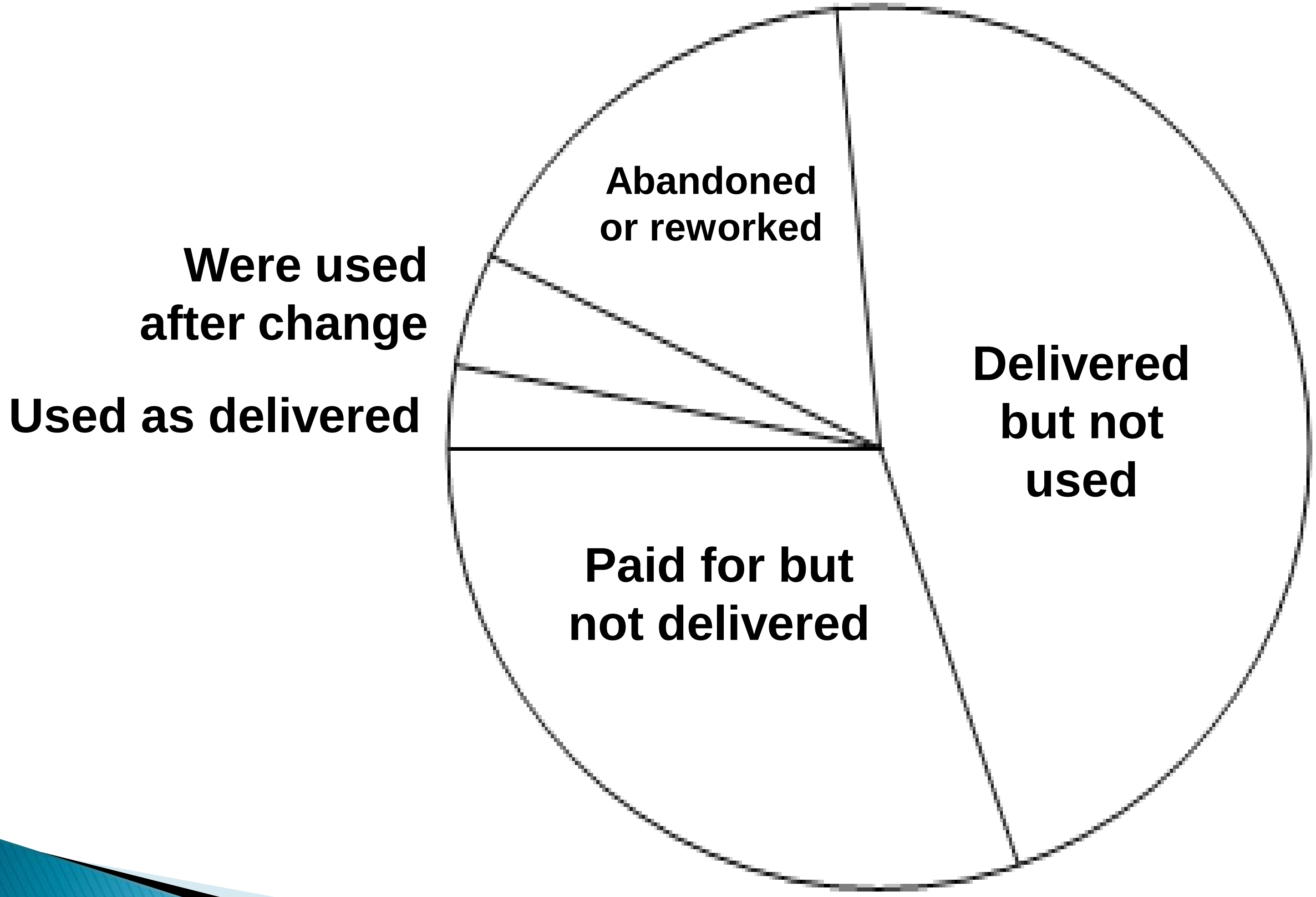


**Our work will be based on Object Oriented Concepts (OOP).**

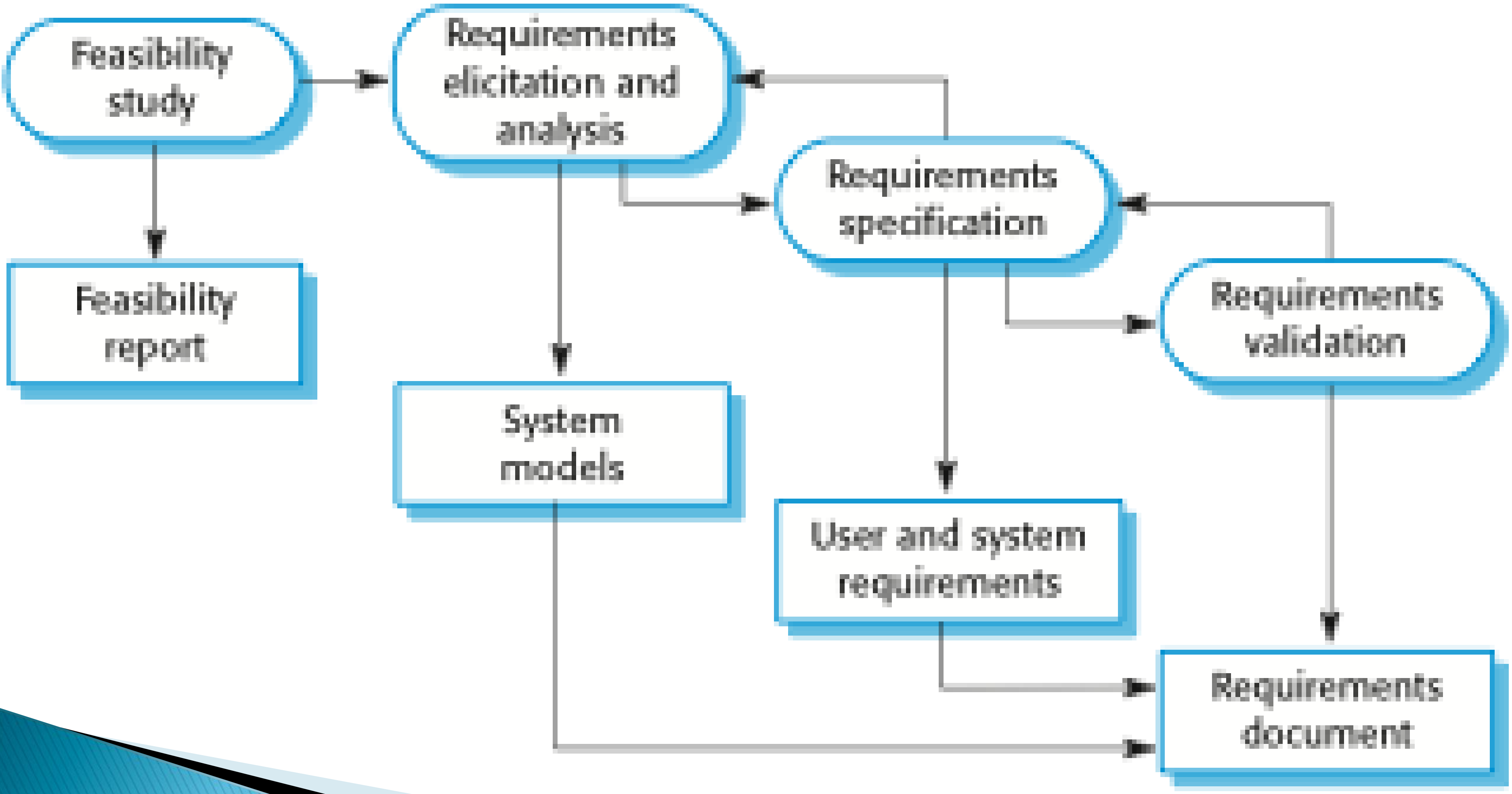
# Approximate relative costs of the phases of SDLC



# Software Implementation Status



# The Requirements Engineering Process



# Software Design and Implementation

- The process of converting the system specification into an executable system.

- Software design

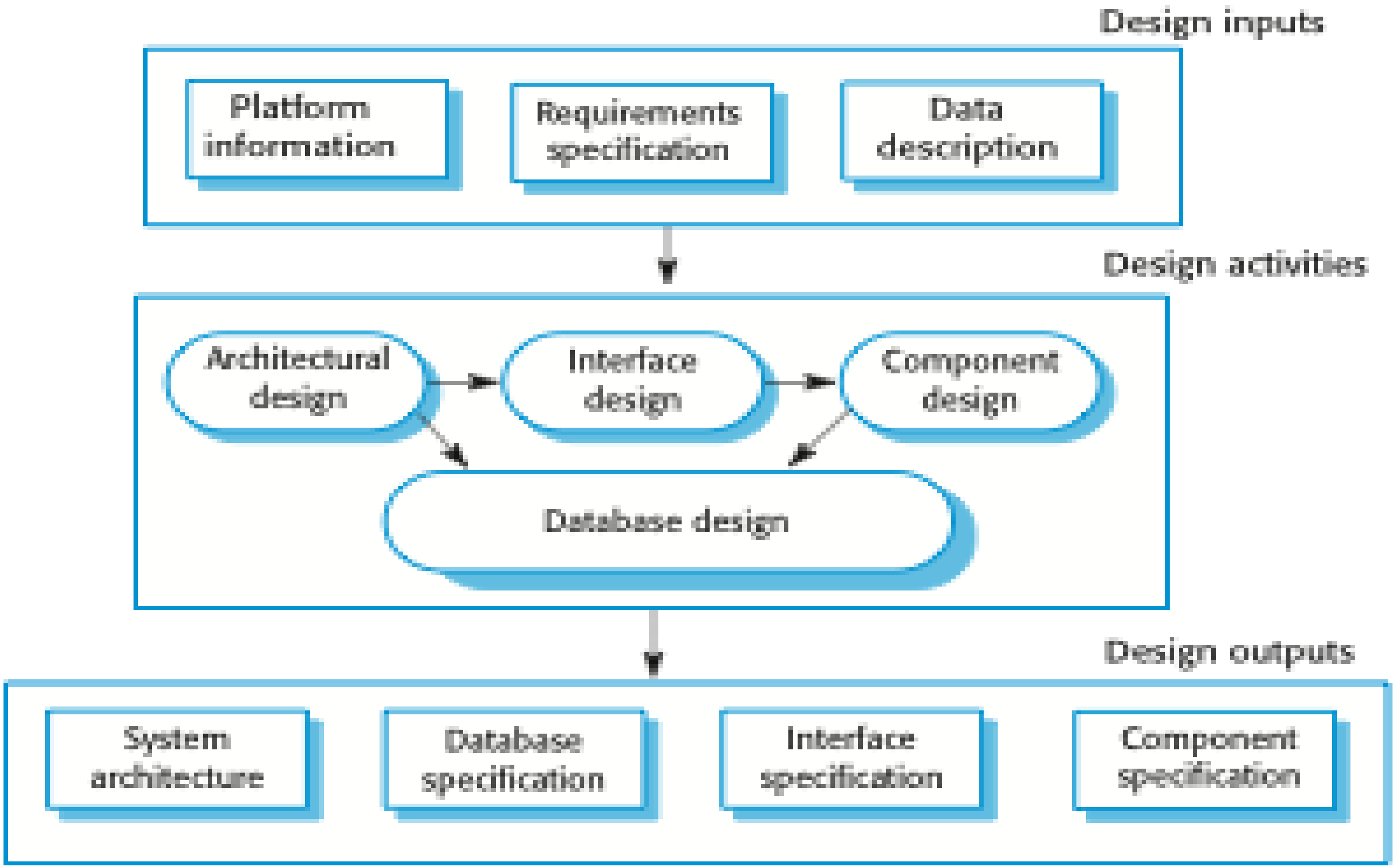
  Design a software structure that realises the specification;

- Implementation

  Translate this structure into an executable program;

- The activities of design and implementation are closely related and may be inter-leaved.

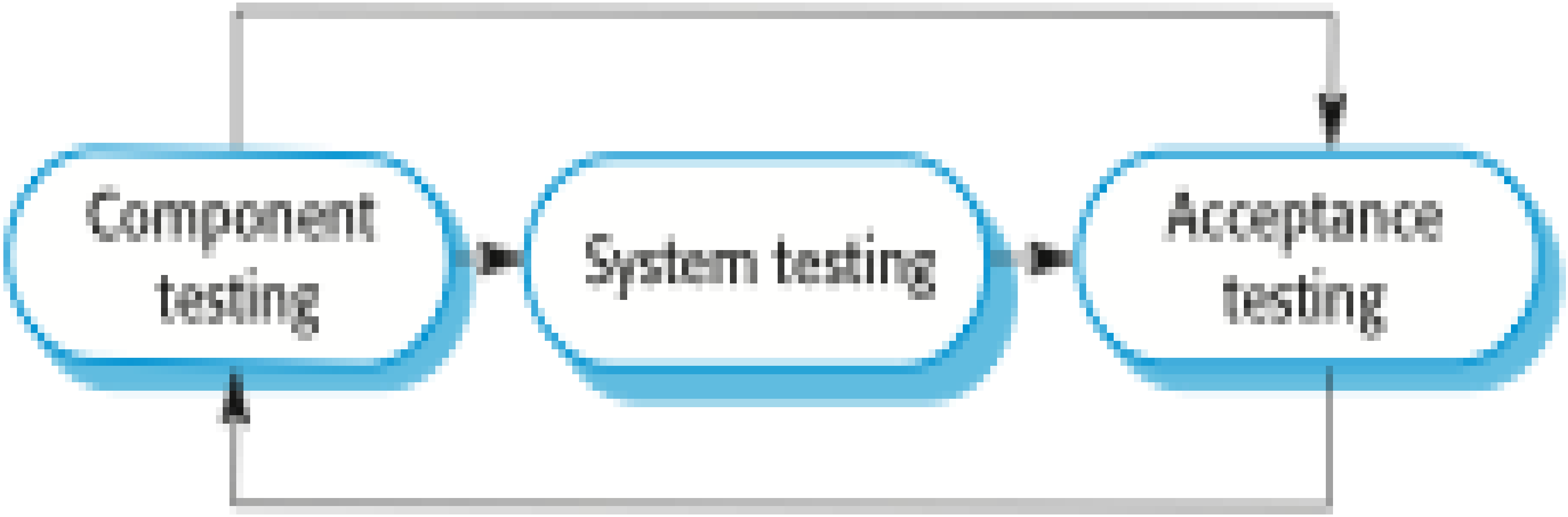
# A General Model of The Design Process



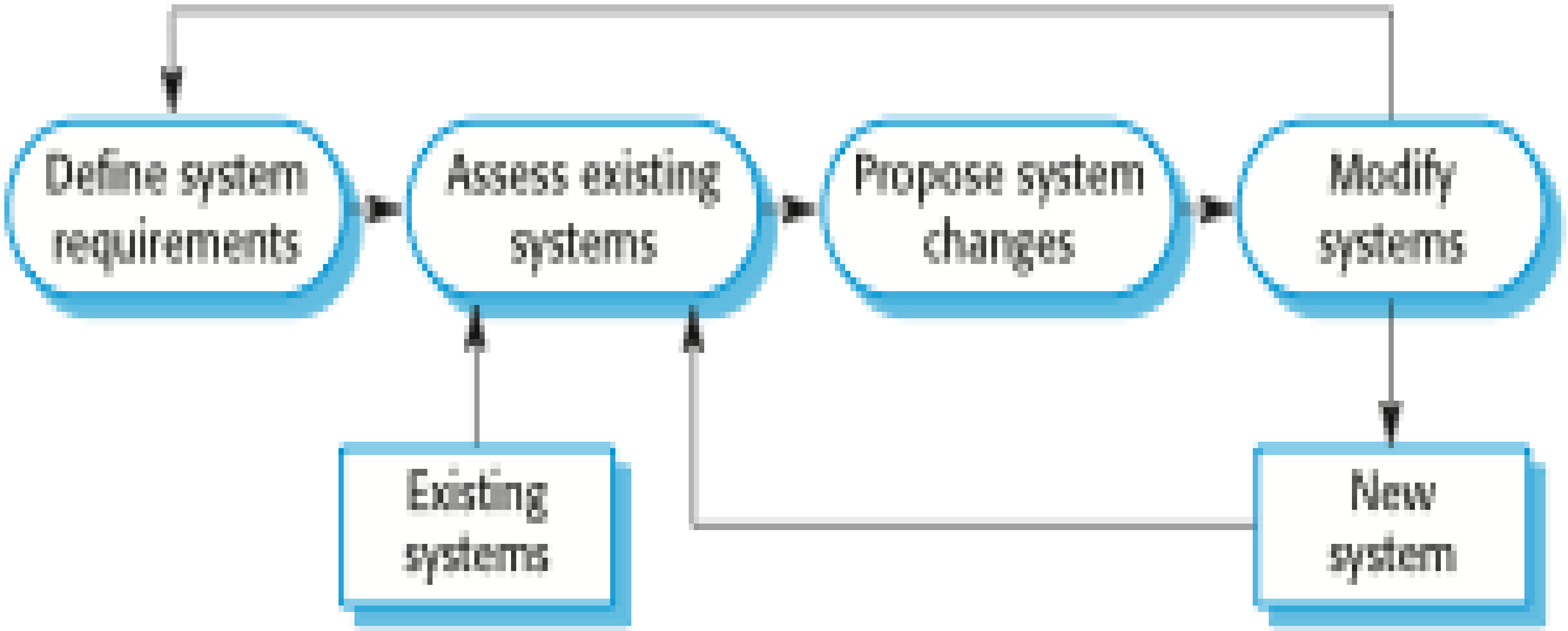
# Software validation

- Verification and validation (V & V) is intended to show that a system conforms to its specification and meets the requirements of the system customer.
- Involves checking and review processes and system testing.
- System testing involves executing the system with test cases that are derived from the specification of the real data to be processed by the system.
- Testing is the most commonly used V & V activity.

# Stages of Testing



# System Evolution



# Questions