Software Process Models

lecture 2

Prepared by:

Dr. Lobna Mohamed Abou El-magd

topics

- Software process models
- Process iteration
- Process activities
- Computer-aided software engineering

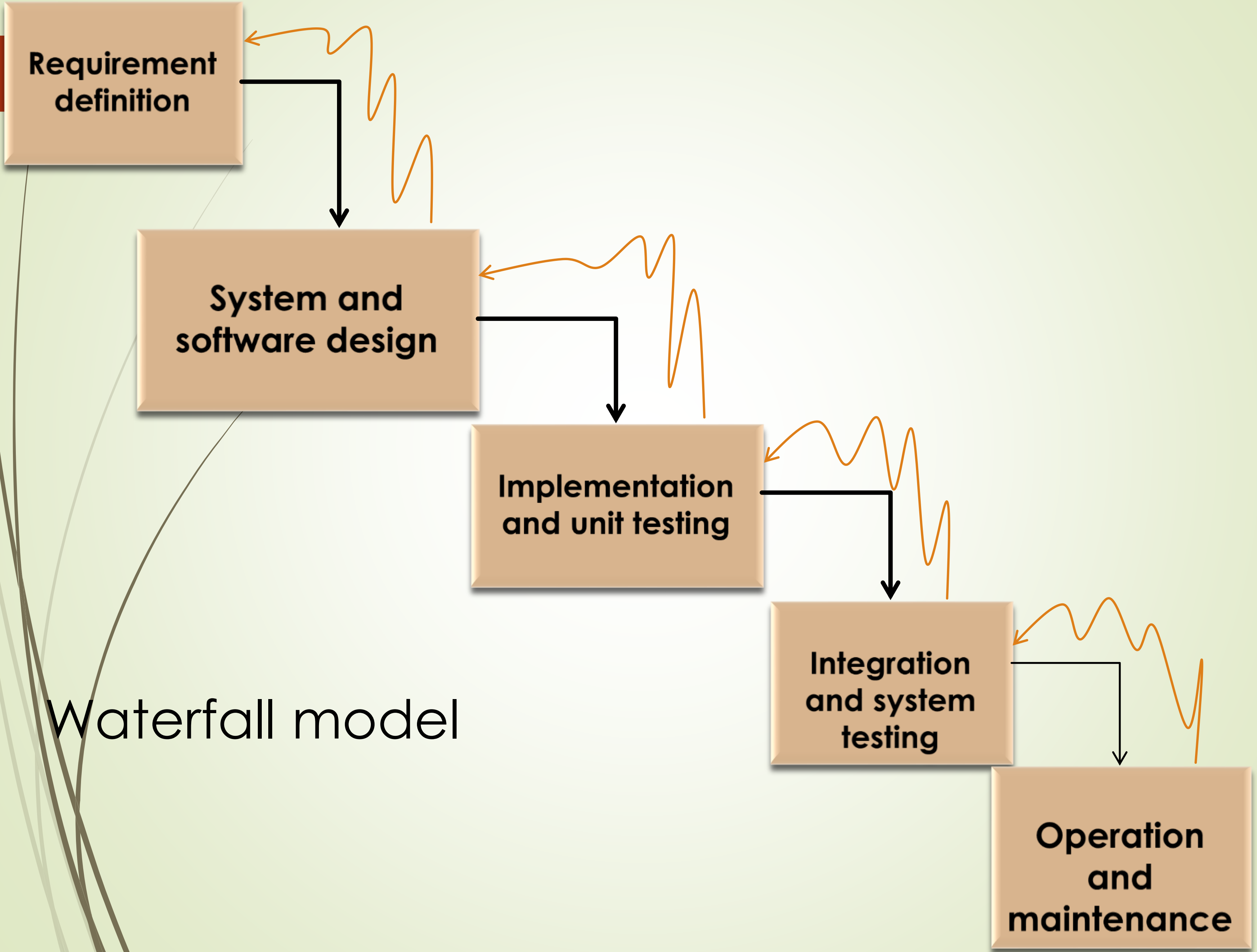
The software process

- A structured set of activities required to develop a software system
  - Specification;
  - Design;
  - Validation;
  - Evolution.
- A software process model is an abstract representation of a process. It presents a description of a process from some particular perspective.

Generic software process models

- The waterfall model
  - Separate and distinct phases of specification and development.
- Evolutionary development
  - Specification, development and validation are interleaved.
- Component-based software engineering
  - The system is assembled from existing components.
- These three generic process models are widely used in current software engineering practice. They are not mutually exclusive and are often used together, especially for large systems development.

Waterfall model


Requirement definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Waterfall model phases

- Requirements analysis and definition
- System and software design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance
- The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway. One phase has to be complete before moving onto the next phase.

Waterfall model problems

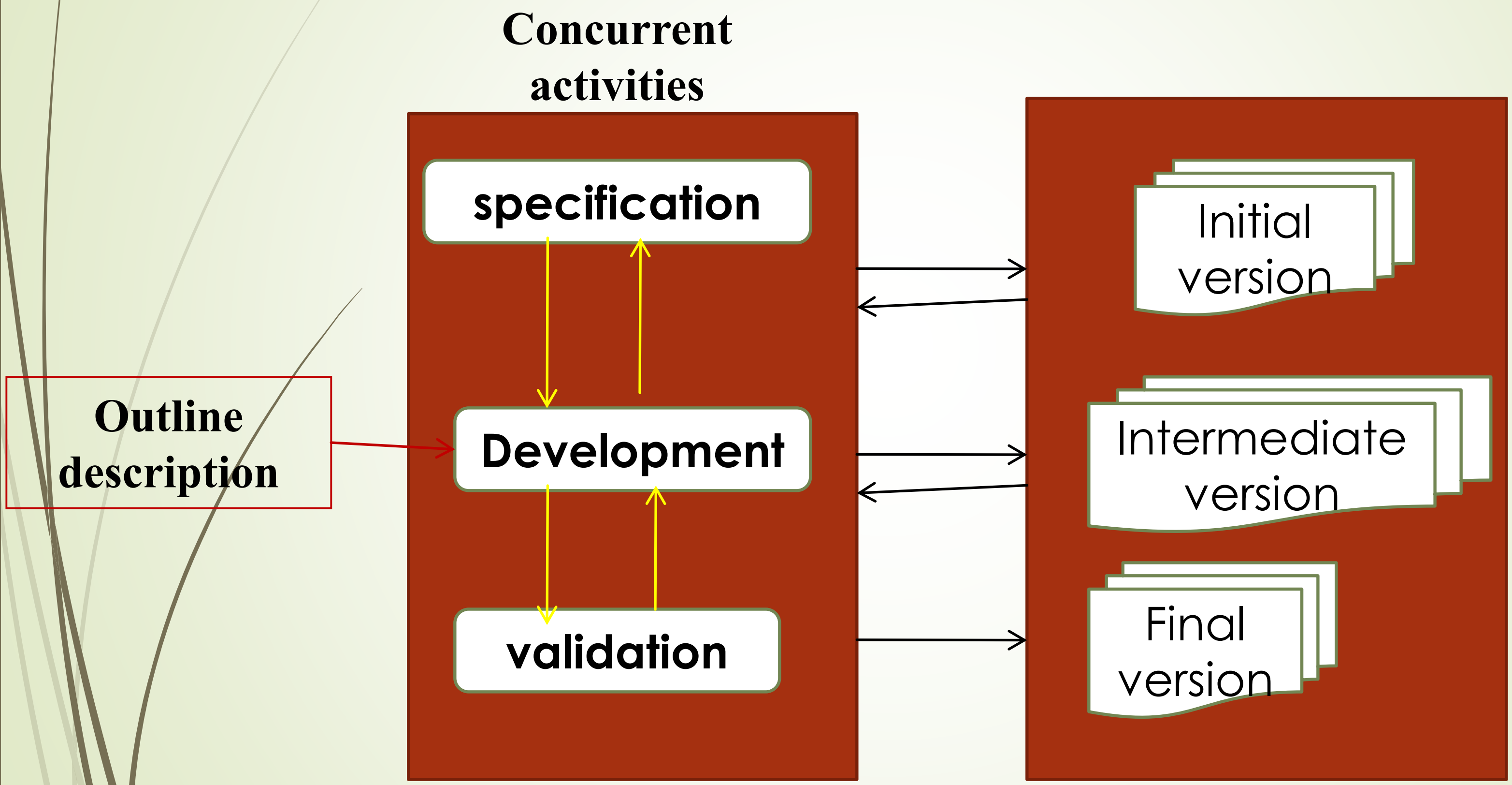
- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
- Therefore, this model is only appropriate when the requirements are well-understood and changes will be limited during the design process.
- Few business systems have stable requirements.
- The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.

Evolutionary development

There are two types of evolutionary development:

- □ **Exploratory development**
  - Objective is to work with customers to explore their requirements and deliver a final system.
  - The development starts with the parts of the system that are understood.
  - the system evolves by adding new features proposed by the customer.
- □ **Throw-away prototyping**
  - Objective is to understand the system requirements.
  - Should start with poorly understood requirements to clarify what is really needed.

Evolutionary development



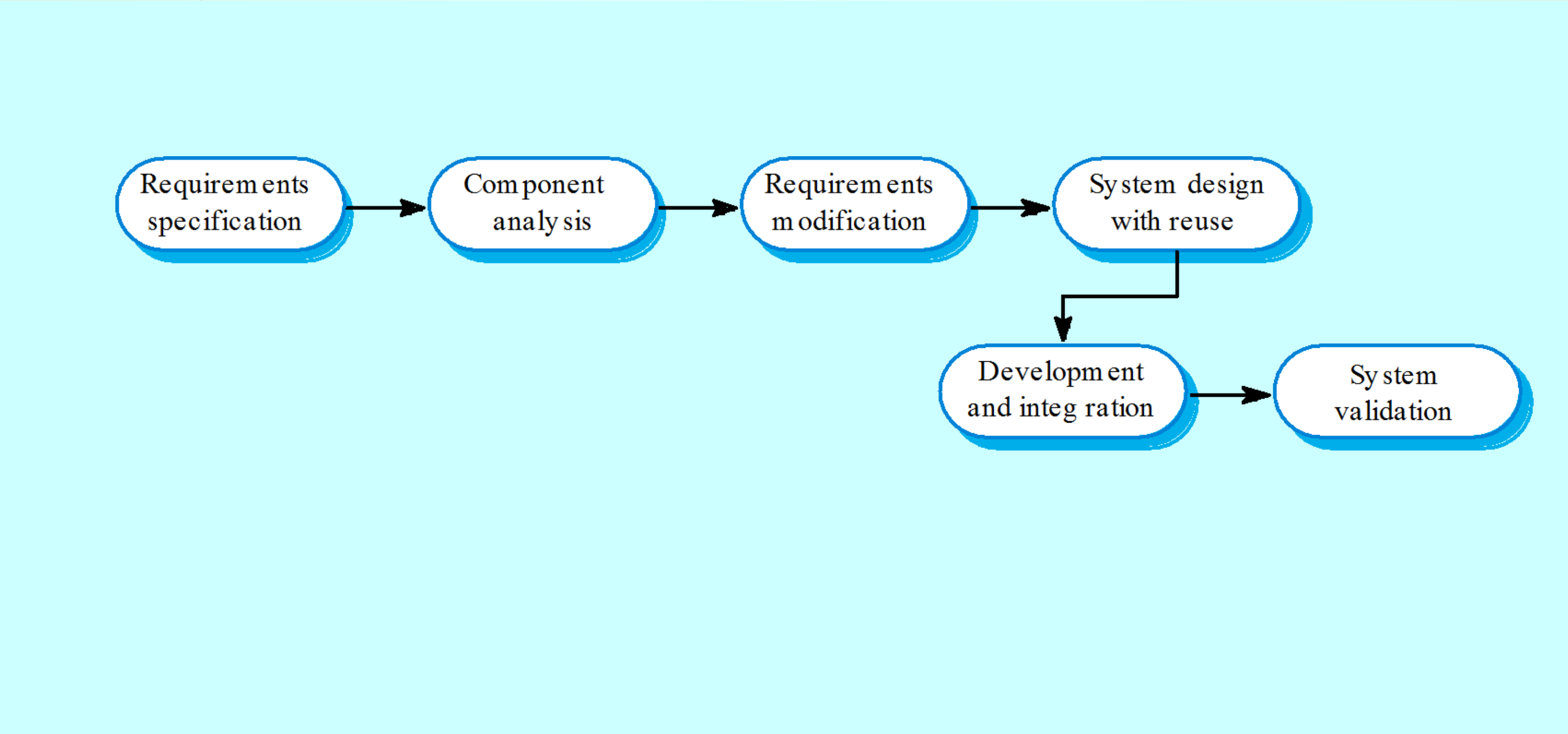
Evolutionary development

- Problems
  - Lack of process visibility;
  - Systems are often poorly structured;
  - Special skills (e.g. in languages for rapid prototyping) may be required.
- Applicability
  - For small or medium-size interactive systems;
  - For parts of large systems (e.g. the user interface);
  - For short-lifetime systems.

Component-based software engineering (CBSE)

- Based on systematic reuse where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.
- Process stages
  - Component analysis;
  - Requirements modification;
  - System design with reuse;
  - Development and integration.
- This approach is becoming increasingly used as component standards have emerged.

Reuse-oriented development



Requirements specification

Component analysis

Requirements modification

System design with reuse

Development and integration

System validation

Process iteration

- The system requirements change as the business procuring the system responds to external pressures. Management priorities change. As new technologies become available, designs and implementation change.
- This means that the software process is not a one-off process; rather, the process activities are regularly repeated as the system is reworked in response to change requests.

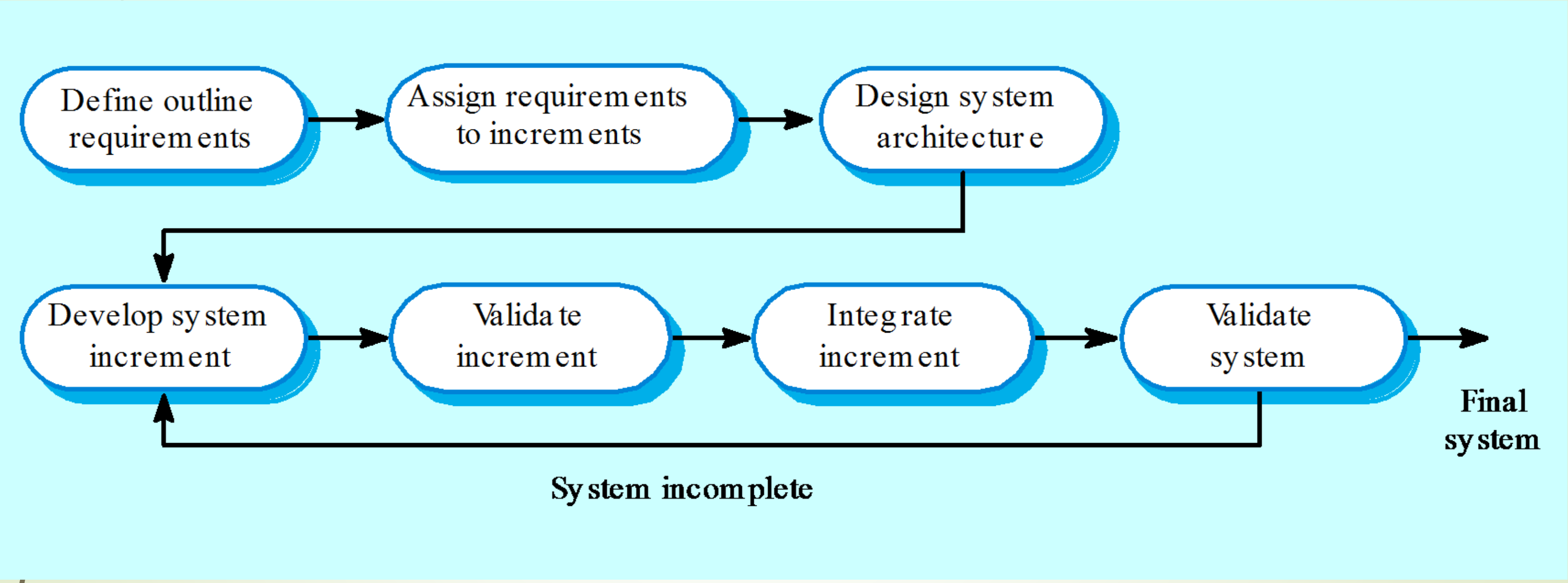
Process iteration

- Iteration can be applied to any of the generic process models.
- Two (related) approaches
  1. Incremental delivery;
  2. Spiral development.

1- Incremental delivery

- Rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments.
- Once the development of an increment is started, the requirements are frozen, but requirements for later increments can continue to evolve.

Incremental development



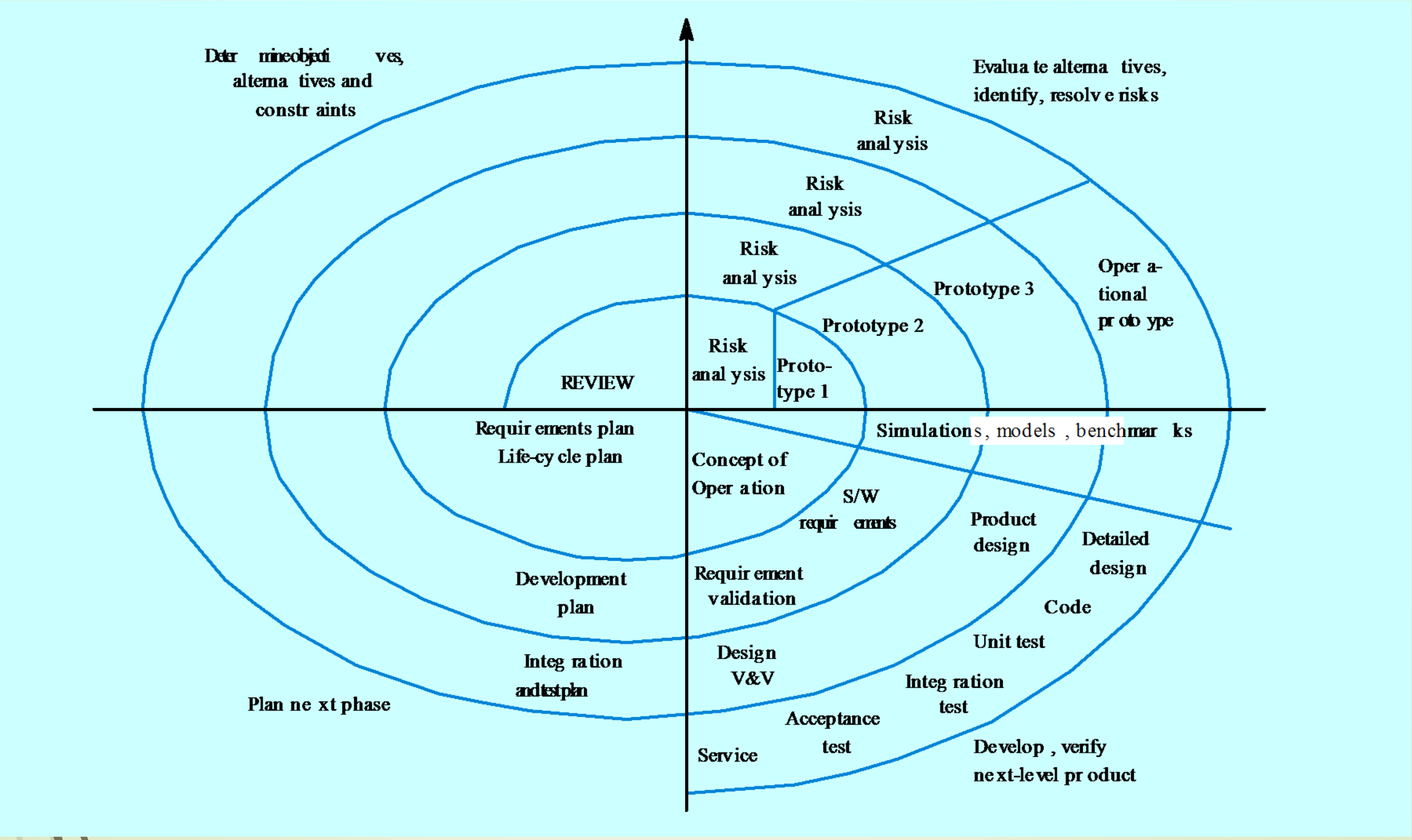
Incremental development advantages

- Customer value can be delivered with each increment so system functionality is available earlier.
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.

2-Spiral development

- Process is represented as a spiral rather than as a sequence of activities with backtracking.
- Each loop in the spiral represents a phase in the process.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
- Risks are explicitly assessed and resolved throughout the process.

Spiral model of the software process



Determine objectives, alternatives and constraints

Evaluate alternatives, identify, resolve risks

Risk analysis

Risk analysis

Risk analysis

Risk analysis

Prototype 3

Prototype 2

Proto-type 1

Operational protoype

REVIEW

Requirements plan Life-cycle plan

Development plan

Integration and test plan

Plan next phase

Simulations, models, benchmarks

Concept of Operation

S/W requirements

Requirement validation

Product design

Detailed design

Code

Unit test

Design V&V

Integration test

Acceptance test

Service

Develop, verify next-level product

Spiral model sectors

- Objective setting
  - Specific objectives for the phase are identified.
- Risk assessment and reduction
  - Risks are assessed and activities put in place to reduce the key risks.
- Development and validation
  - A development model for the system is chosen which can be any of the generic models.
- Planning
  - The project is reviewed and the next phase of the spiral is planned.

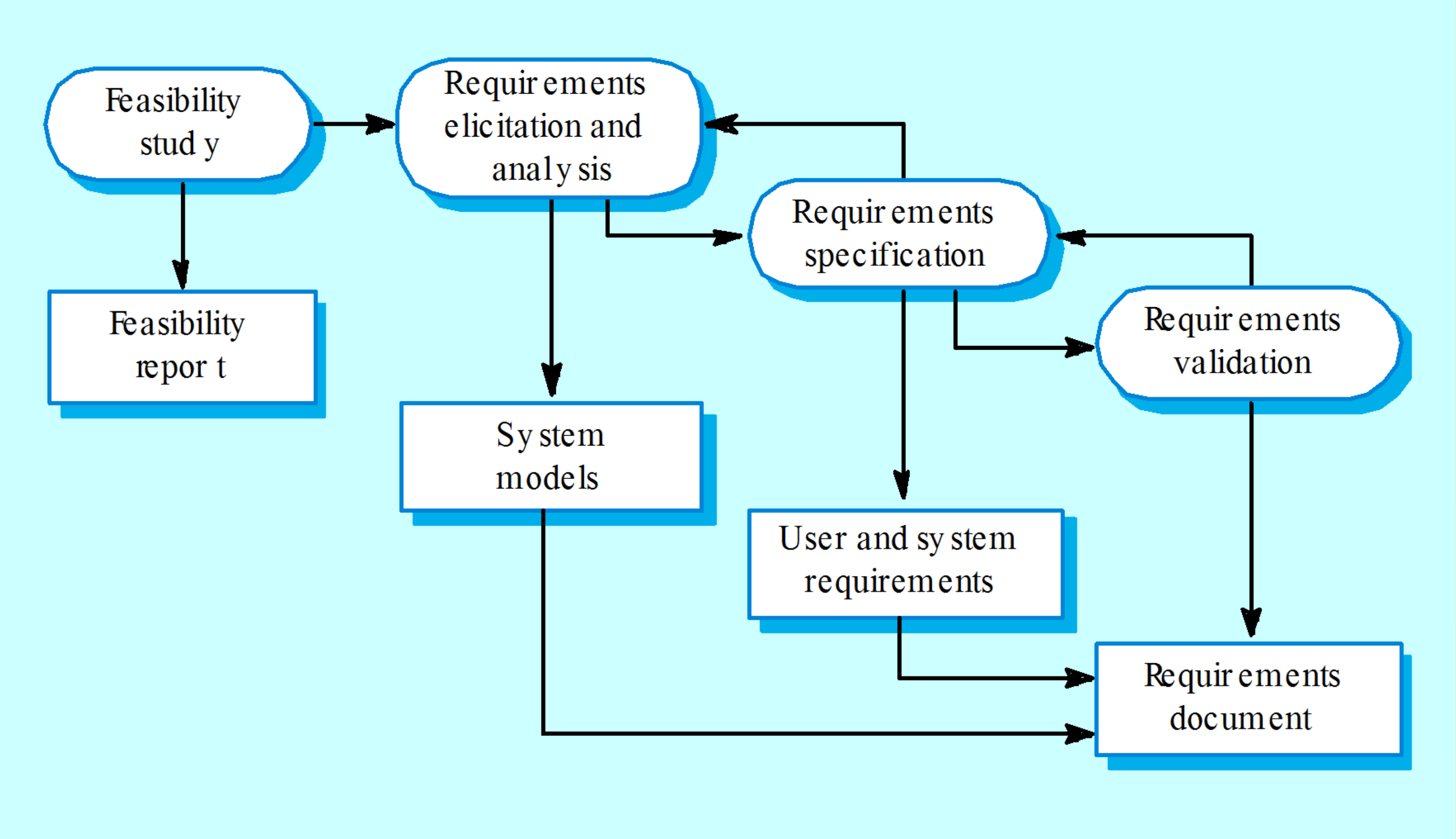
Process activities

1. Software specification
2. Software design and implementation
3. Software validation
4. Software evolution

1-Software specification

- The process of establishing what services are required and the constraints on the system’s operation and development.
- There are four main phases in the requirements engineering process:
  - Feasibility study;
  - Requirements elicitation and analysis;
  - Requirements specification;(Two types of requirements *User requirements system requirements)*
  - Requirements validation. (checks the requirements for realism, consistency and completeness)

The requirements engineering process



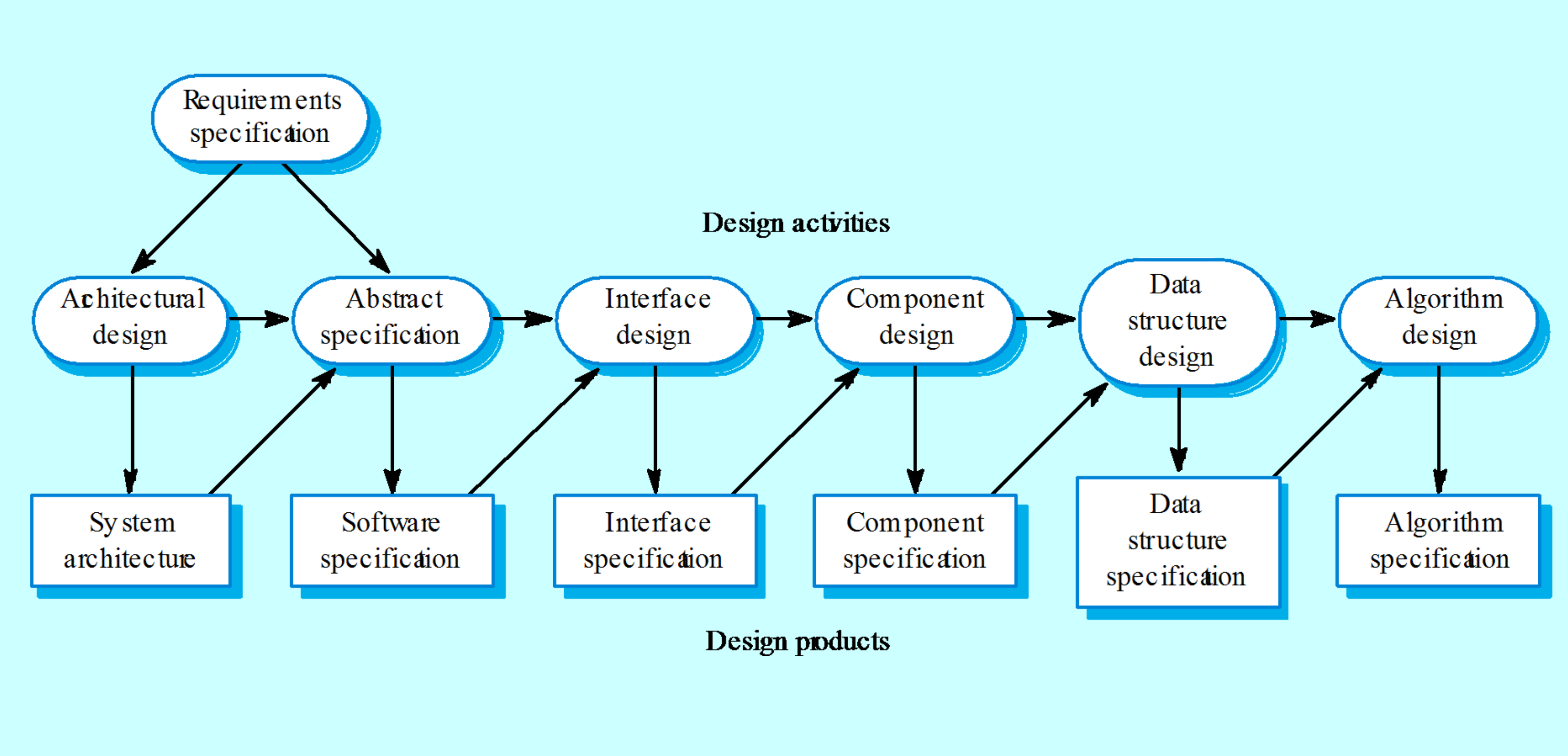
2-Software design and implementation

- 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.

Design process activities

- Architectural design
- Abstract specification
- Interface design
- Component design
- Data structure design
- Algorithm design

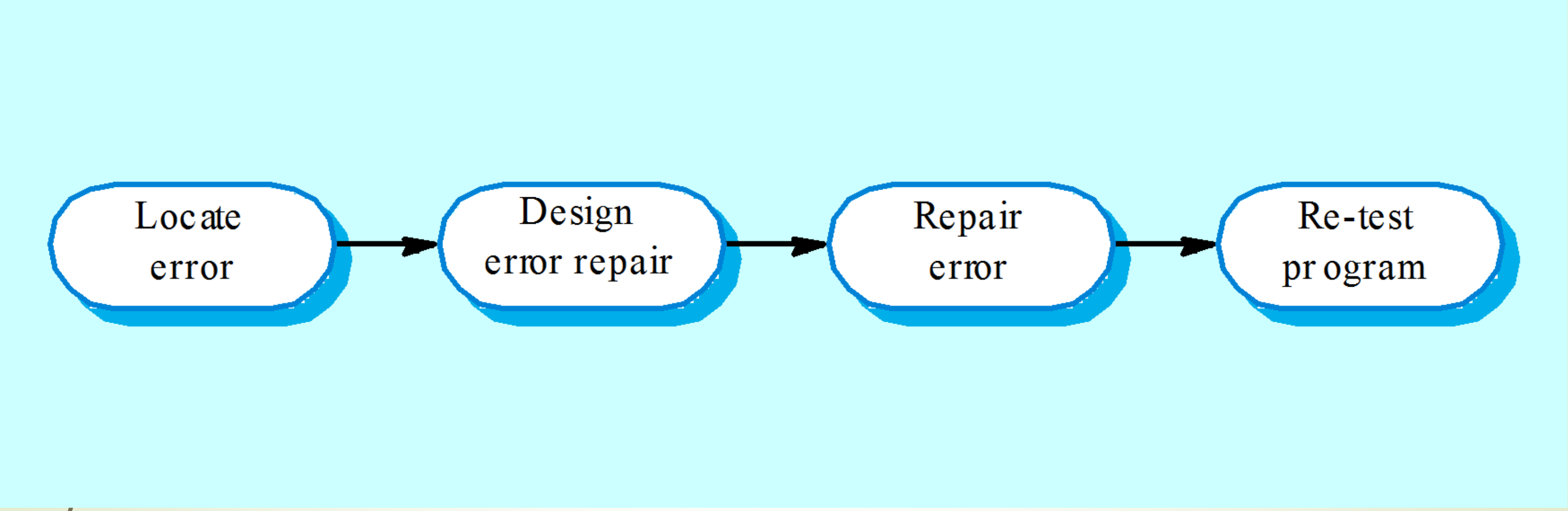
The software design process



Programming and debugging

- Translating a design into a program and removing errors from that program.
- Programming is a personal activity - there is no generic programming process.
- Programmers carry out some program testing to discover faults in the program and remove these faults in the debugging process.

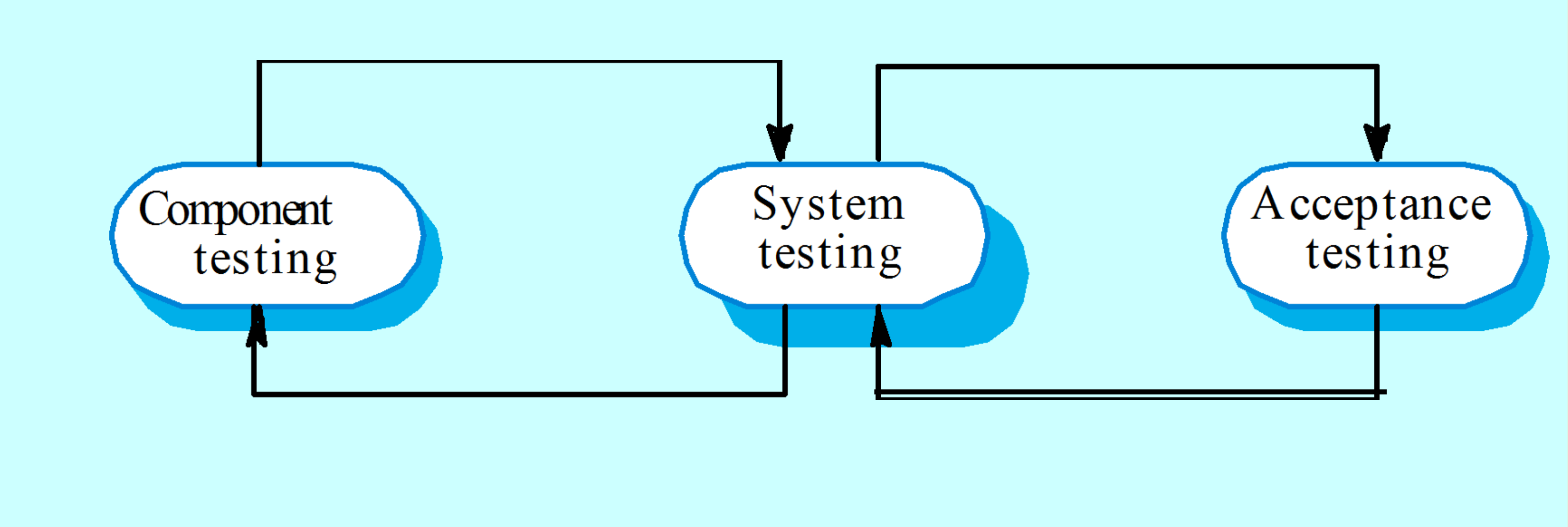
The debugging process



3- 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.

The testing process



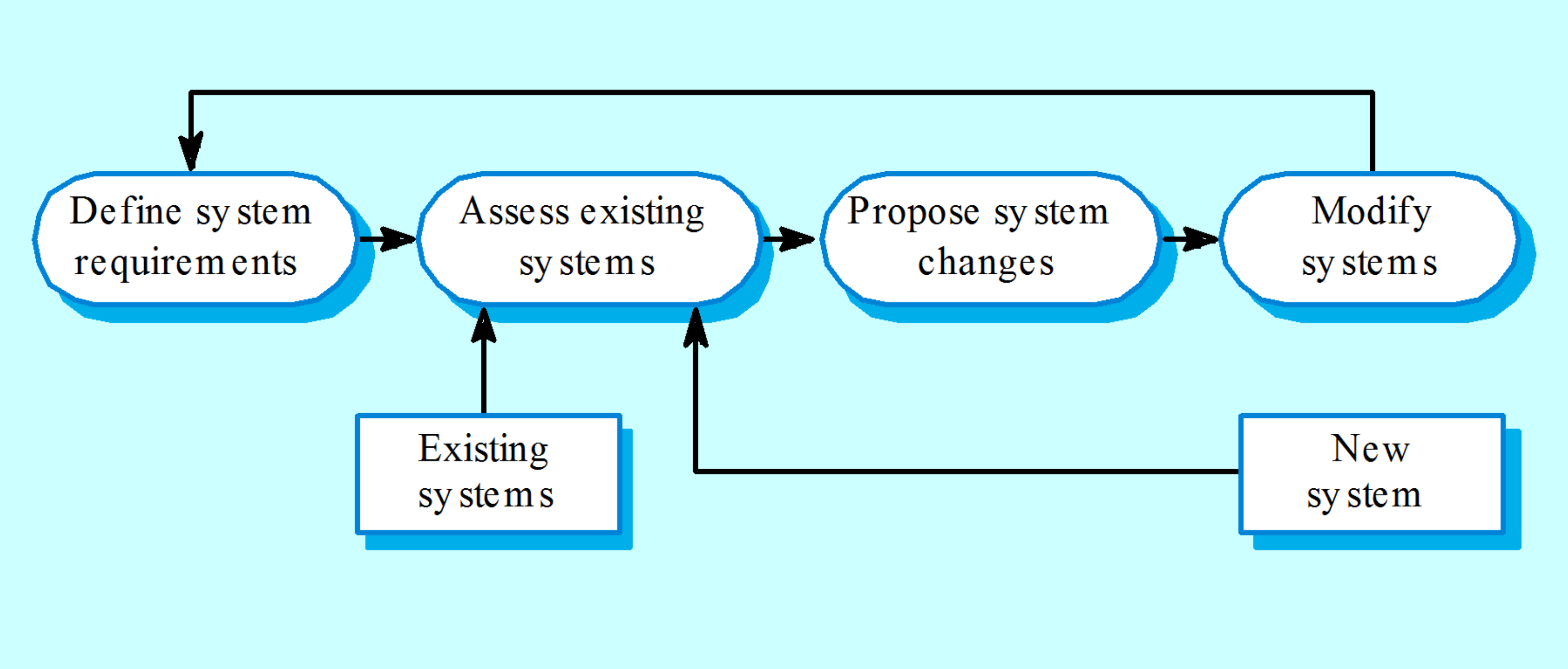
Testing stages

- Component or unit testing
  - Individual components are tested independently;
  - Components may be functions or objects or coherent groupings of these entities.
- System testing
  - Testing of the system as a whole. Testing of emergent properties is particularly important.
- Acceptance testing
  - Testing with customer data to check that the system meets the customer’s needs.

4-Software evolution

- Software is inherently flexible and can change.
- As requirements change through changing business circumstances, the software that supports the business must also evolve and change.

System evolution



Computer-aided software engineering

- Computer-aided software engineering (CASE) is software to support software development and evolution processes.
- Activity automation
  - Graphical editors for system model development;
  - Data dictionary to manage design entities;
  - Graphical UI builder for user interface construction;
  - Debuggers to support program fault finding;
  - Automated translators to generate new versions of a program.

Questions?

1- state the process activities.

2- compare between different process model.b